Department of Information and Computer Science

Real-time Action Recognition for RGB-D and Motion Capture Data

Xi Chen

Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 207/2014

Real-time Action Recognition for RGB-D and Motion Capture Data

Xi Chen

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the lecture hall T2 of the school on 16 January 2015 at 12.

Aalto University
School of Science
Department of Information and Computer Science

Supervising professor
Aalto Distinguished Professor Erkki Oja

Thesis advisor
Dr. Markus Koskela

Preliminary examiners
Professor Guoying Zhao, University of Oulu, Finland
Professor Vassilis Athitsos, University of Texas at Arlington, United States

Opponent
INRIA research director, Dr. Ivan Laptev, France

Aalto University publication series
DOCTORAL DISSERTATIONS 207/2014

ISBN 978-952-60-6013-2 (printed)
ISBN 978-952-60-6014-9 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-6014-9

Images: Taru Falenius

Unigrafia Oy
Helsinki 2014

Finland

441 697
Printed matter

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Xi Chen

Name of the doctoral dissertation
Real-time Action Recognition for RGB-D and Motion Capture Data

Publisher School of Science

Unit Department of Information and Computer Science

Series Aalto University publication series DOCTORAL DISSERTATIONS 207/2014

Field of research Information and Computer Science

Manuscript submitted 1 September 2014 **Date of the defence** 16 January 2015

Permission to publish granted (date) 28 October 2014 **Language** English

☐ **Monograph** ☒ **Article dissertation (summary + original articles)**

Abstract

In daily life humans perform a great number of actions continuously. We recognize and interpret these actions unconsciously while interacting and communicating with people and the environment. If the machines and computers could also recognize human gestures as effectively as human beings, a new world would be unfolded, filled with a large number of applications to facilitate our daily life. These significant benefits for the society have motivated the research on machine-based gesture recognition, which has already shown some initial advantages in many applications. For example, gestures can be used as commands to control robots or computer programs instead of using standard input devices such as touch screens or mice.

This thesis proposes a framework for gesture recognition systems based on motion capture and RGB-D data. Motion capture data consists of positions and orientations of the key joints of the human skeleton. RGB-D data contains the RGB image and depth data from which a skeletal model can be learnt. This skeletal model can be seen as a noisy approximation of the more accurate motion capture skeleton model. The modular design of our framework enables convenient recognition using multiple data modalities.

The first part of the thesis introduces various methods used in existing recognition systems in the literature and a brief introduction of the proposed real-time recognition system for both whole body gestures and hand gestures. The second part of the thesis is a collection of eight publications by the author of the thesis. Detailed information about the proposed recognition system can be found in these publications. In general, the framework can be roughly divided into two parts, feature extraction and classification. Both have significant influence on the recognition performance. Multiple features are developed and extracted from the skeletons, images, and depth data for each frame in the motion sequence. These features are combined in the early fusion stage, and classified by a single hidden layer neural network - extreme learning machine. The frame-level classification outputs are then aggregated on the sequence level to obtain the final classification result.

The methodologies used in the gesture recognition system are also applied in a proposed image retrieval system. Several image features are extracted and search algorithms are applied to achieve a fast and accurate retrieval. Furthermore, a method is also proposed to align different motion sequences and to evaluate the alignment. The method can be used for gesture retrieval and for skeleton generation algorithm evaluation.

Keywords Action recognition, gesture recognition, RGB-D, motion capture, extreme learning machine, computer vision, machine learning, image retrieval

ISBN (printed) 978-952-60-6013-2 **ISBN (pdf)** 978-952-60-6014-9

ISSN-L 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2014

Pages 191 **urn** http://urn.fi/URN:ISBN:978-952-60-6014-9

Preface

The work presented in this thesis has been carried out in the Department of Information and Computer Science in Aalto University School of Science (former Helsinki University of Technology) under the supervision of Professor Erkki Oja and Dr. Markus Koskela. The work has been funded by multimodal augmented reality (UI-ART) project of the Aalto MIDE programme, Finnish Center of Excellence in Computational Inference Research (COIN), and Academy of Finland Project Multimodally Grounded Language Technology. The Finnish Doctoral Programme in Computational Sciences (FICS) has generously funded my travelling to multiple conferences. Without the significant support from my supervisors, the department, and Aalto University, this work would not have been possible. Words cannot express my appreciation, but let me try: thank you, thank you and thank you.

I am sincerely thankful to my supervisor, Professor Erkki Oja. Thank you for the invaluable guidance, encouragement and financial support throughout my research. I would like to thank my instructor Dr. Markus Koskela, who is always available to help and discuss, from the Linux commands, LaTeXtricks to improving my writing word by word. For the years I have been a part of the content-based image and information retrieval (CBIR) group, I would like to thank Docent Jorma Laaksonen, who gave me the opportunity to work in this group and financed me through a couple of projects. I would also like to thank other current and previous group members, Dr. Ville Viitaniemi, Mats Sjöberg and Dr. He Zhang.

It has been a pleasure to work in Professor Timo Honkela's project. Thanks for all the discussions and support. I would like to thank Docent Zhirong Yang for his fruitful advice. During my years as a doctoral candidate at the department, I have been lucky to collaborate and discuss with many researchers. I would like to thank Kyunghyun Cho, Jouko

Hyväkkä of VTT, Klaus Förger, Mark van Heeswijk, Dušan Sovilj, Oskar Kohonen, Yoan Miche and Professor Amaury Lendasse of University of Iowa. Besides, I want to thank the secretaries Minna Kauppila, Leila Koivisto and Tarja Pihamaa for arranging my conference trips and other practical issues of the department during these years. Unfortunately, due to the space restriction I can not list all the colleagues, but I would like to thank all the others from the department as well. Thank you!

Especially I would like to thank Professor Guangbin Huang of Nanyang Technology University, Singapore, whom I first met during his research visit in our department. As the inventor of the extreme learning machine, his algorithm has significantly promoted my research work. And his words have inspired and encouraged me in my research.

I express my gratitude to INRIA research director Ivan Laptev, the opponent in my defense. I would like to thank the pre-examiners of the dissertation: Professor Guoying Zhao of University of Oulu, Finland and Professor Vassilis Athitsos of University of Texas at Arlington, United States for their valuable and thorough comments on the dissertation.

During these years in Finland, I have met many friends. Some of them are from ICS and CSE departments, with whom I have had lunch in school for years. Thank you for your company. The rest I can not list all of them, but let me try to thank at least a few: Yong Han, Mark Sevalnev, Zengcai Qu, Tele Hao, Xiaopeng Hong, Pentti and Marja.

I am extremely grateful to my parents for their enormous support throughout my studies. And I would like to thank my parents-in-law and sister-in-law for all the happiness and care that they have brought to me. Most of all, I wish to thank my husband Kari, who has enriched my life with his wide knowledge and wisdom. His love and support has encouraged me to accomplish this thesis.

Espoo, November 26, 2014,

Xi Chen

Contents

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Xi Chen and Markus Koskela and Jouko Hyväkkä. Image Based Information Access for Mobile Phones. In *Proceedings of 8th International Workshop on Content-Based Multimedia Indexing (CBMI2010)*, pages 1-5, Grenoble, France, June 2010.

II Xi Chen and Markus Koskela. Mobile Visual Search from Dynamic Image Databases. In *Proceedings of 17th Scandinavian Conference on Image Analysis (SCIA 2011)*, pages 196-205, Ystad, Sweden, May 2011.

III Xi Chen and Markus Koskela. Classification of RGB-D and Motion Capture Sequences Using Extreme Learning Machine. In *Proceedings of 18th Scandinavian Conference on Image Analysis (SCIA 2013)*, pages 640-651, Espoo, Finland, June 2013.

IV Xi Chen and Markus Koskela. Skeleton-Based Action Recognition with Extreme Learning Machines. *Neurocomputing*, Volume 149, Part A, Pages 387-396, February 2015.

V Xi Chen and Markus Koskela. Sequence Alignment for RGB-D and Motion Capture Skeletons. In *Proceedings of the International Conference on Image Analysis and Recognition (ICIAR 2013)*, pages 630-639, Póvoa de Varzim, Portugal, June 2013.

VI Kyunghyun Cho and Xi Chen. Classifying and Visualizing Motion Capture Sequences using Deep Neural Networks. In *Proceedings of the 9th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 122-130, Lisbon, Portugal, January 2014.

VII Xi Chen and Markus Koskela. Online RGB-D Gesture Recognition with Extreme Learning Machines. In *Proceedings of the 15th ACM International Conference on Multimodal Interaction (ICMI 2013)*, pages 467-474, Sydney, Australia, December 2013.

VIII Xi Chen and Markus Koskela. Using Appearance-Based Hand Features For Dynamic RGB-D Gesture Recognition. In *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR14)*, pages 411-416, Stockholm, Sweden, August 2014.

Author's Contribution

Publication I: "Image Based Information Access for Mobile Phones"

The author proposed the prototype of the image matching system and implemented it by Matlab.

Publication II: "Mobile Visual Search from Dynamic Image Databases"

The author proposed the dynamic image database and the descriptor pruning method to reduce memory cost, and conducted the experiments on the magazine database.

Publication III: "Classification of RGB-D and Motion Capture Sequences Using Extreme Learning Machine"

The author proposed the features and the classification framework, and conducted the experiments. The recording of Kinect skeleton database was a collaborative effort.

Publication IV: "Skeleton-Based Action Recognition with Extreme Learning Machines"

The author proposed the action recognition system and conducted the experiments.

Publication V: "Sequence Alignment for RGB-D and Motion Capture Skeletons"

The author proposed using the dynamic time warping with a varied step size to align the skeletons and the evaluation methods, and conducted experiments for the alignment. The recording of Kinect skeletons and motion capture skeletons was a collaborative effort.

Publication VI: "Classifying and Visualizing Motion Capture Sequences using Deep Neural Networks"

The author extracted the features, proposed the gesture recognition system, and conducted part of the experiments. Both authors contributed equally in this work.

Publication VII: "Online RGB-D Gesture Recognition with Extreme Learning Machines"

The recognition framework was designed together. The author had a major role in conducting the experiments.

Publication VIII: "Using Appearance-Based Hand Features For Dynamic RGB-D Gesture Recognition"

The recognition framework was designed together. The author had a major role in conducting the experiments.

Abbreviations

ANN	approximate nearest neighbors
ASF/AMC	acclaim skeleton file/acclaim motion capture
ASL	American sign language
BoVW	bag of visual words
BVH	biovision hierarchy
C3D	coordinate 3D
CBIR	content based image/information retrieval
CCD	comparative coding descriptors
CMU	Carnegie Mellon University
CRF	conditional random field
DCSF	depth cuboid similarity feature
DMM	depth motion map
DoG	difference of Gaussians
DSTIPs	spatio-temporal interesting points from depth videos
DTW	dynamic time warping
ELM	extreme learning machine
FLANN	fast library for approximate nearest neighbors
FPS	frames per second
GMM	Gaussian mixture model
HCI	human computer interaction
HMM	hidden Markov model
HOF	histogram of optical flow
HOG	histogram of oriented gradients
HOG3D	histogram of oriented 3D spatio-temporal gradients
HON4D	histogram of oriented 4D normals
IR	infrared
KNN	K-nearest neighbors
LoG	Laplacian of Gaussian

MAP	maximum a posteriori
MHAD	Multimodal Human Action Database
MOCAP	motion capture
MSER	maximally stable extremal region
MSR	Microsoft Research
MSRC	Microsoft Research Cambridge
NiTE	natural interaction middleware
PCA	principal component analysis
QR	quick response
RBF	radial basis function
RGB	red-green-blue
RGB-D	RGB and depth
ROP	random occupancy pattern
SDK	software development kit
SIFT	scale-invariant feature transform
SLFN	single-hidden layer feed-forward neural network
SOM	self-organising map
STIP	spatio-temporal interesting points
SURF	speed-up robust features
SVD	singular value decomposition
SVM	support vector machine
TUM	Technische Universität München
VGA	video graphics array

1. Introduction

1.1 Background

Humans perform various actions during normal daily life. We eat, walk, nod our heads while engaging in a conversation, and so on. All these actions enable us to continuously interact with our surroundings, the environment and people. On the other hand, we also perceive these actions performed by others and respond accordingly.

Due to the rapid development of computer vision and machine learning, recognizing human actions through image or motion sensors has gained increasing popularity. Consequently, this enables multiple possibilities in a large number of applications, such as surveillance, analysis of sign language, human–computer interaction (HCI), gaming, and robotic control. For example, we can use body or hand gestures to control robots to execute desired tasks as long as the robots can recognize the human gestures correctly using their installed sensors [80, 110], games are developed for deaf children that are based on recognizing American sign language [10], or we can interact with a computer without physical devices by recognizing hand gestures [153]. Numerous applications can benefit from the recognition of human gestures and, therefore, this topic has gained more and more interest in the research community.

A few years ago, the sensors used for the recognition of human gestures were mainly motion capture (mocap) systems and cameras for RGB video. The former generate human skeleton models with 3D joint coordinates based on markers and cameras in a predefined setup [95]. Intensive research has also been conducted for gesture recognition from standard image or video data [31, 132]. However, during the recent few years, the revolutionary low-cost RGB-D (RGB and depth) camera sensors such as

the Microsoft Kinect and Asus Xtion PRO have prevailed in consumer markets. They provide depth information along with the standard RGB video, and due to their low cost are currently widely used e.g. in gaming, HCI, and robotics. Because of the extra depth information, the RGB-D sensor provides more possibilities for recognizing the human actions than regular RGB cameras.

There are many difficulties in designing an effective and robust system to recognize human actions. The system should be independent of the identity of the performers of the actions and the speed of the performance. It should manage the interclass similarity between different actions and intraclass variety of different instances of the same actions. For example, a "kick" action performed by different actors can have different styles (e.g. some might kick in front and some might kick sideways), and even the same actor does not perform the action exactly the same each time. The system should be able to recognize a large number of actions with high recognition accuracy, and in many applications the recognition should be done in real-time.

Action recognition is also denoted as gesture recognition or activity recognition. Strictly speaking, there are very small differences between them. Action recognition emphasizes the movement of the whole body, for example, "walk", "jump", or "sit down". Gesture recognition is more related with the movement of hands or with certain poses of the hands, such as "wave hand" or some sign language gesture. Activity recognition highlights more a full series of actions which are usually more complex and involve several people [149]. For example, "cooking" and "two persons shaking hands" are two typical activities. Anyhow, there are no strict boundaries between these terms. In this thesis, we ignore the little differences between them, and mostly use the terms action and gesture recognition interchangeably.

1.2 Contributions of the thesis

The main theme of this thesis is gesture recognition, which is a wide and well-studied research topic. The content of the thesis is roughly described in Figure 1.1. The data sources for gesture recognition have largely expanded in recent years. Traditionally, the RGB or greyscale video and skeleton data from motion capture are widely used for gesture recognition. Recently, due to the prevalence of low cost RGB-D sensors, the depth data and even the audio data, together with the RGB video

and skeleton data are available for gesture recognition. Furthermore, the acceleration data, which is feasibly available from smart mobile phones, is also used for gesture recognition.

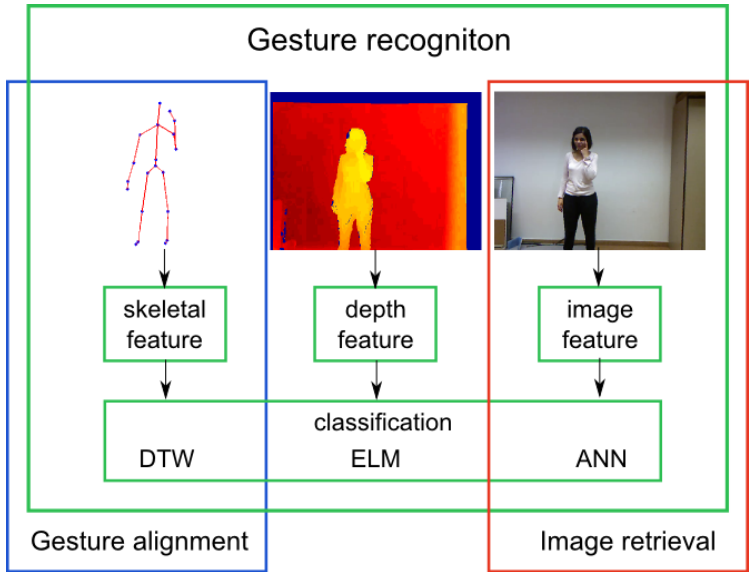


Figure 1.1. Contents of the thesis.

This thesis focuses on dynamic gesture recognition for motion capture and RGB-D sensors, the latter consisting RGB, depth and skeleton data. A dynamic gesture, opposite to a static gesture, consists of a sequence of frames that describe a movement of the body. It is time series data. Consequently, the recognition systems are often designed based on the statistical modeling methods, or time series analysis methods, such as dynamic time warping (DTW). Dynamic time warping measures the similarity between two data sequences. In order to recognize the gestures, it needs to balance between the number of training gesture sequences and the representation coverage of each class. And it is difficult to handle within gesture variance, either due to the different ways of performance or even noise. The statistical methods, such as Hidden Markov Models (HMM), Conditional Random Field (CRF) and the recent popular action graph, require a large amount of training samples in order to train an accurate model, and the training is expensive, both in terms of memory and computing time. In addition, many parameters need to be selected by the developer during the training, which increases the difficulties in training and results in the differences in performance. Moreover, the input of these methods is typically quantized

into discrete integer values. Through quantization, the feature vectors lose certain information embedded in the original features.

Another strategy for dynamic gesture recognition is similar to the static gesture recognition. By considering the whole gesture as a whole and extracting a single feature for the whole gesture sequence, the standard pattern recognition methods can often be applied, for example, Support Vector Machine (SVM). In order to generate a fixed dimensional feature from gestures with different lengths of sequences, two approaches are usually utilized. One approach is to reduce the length of the sequence or resample the sequences, then concatenate the features from each frame in the sequence together. Histogram of codewords is another popular approach. However, the codeword generation, typically implemented by clustering, inevitably leads to quantization loss.

Different from these systems, we develop a novel gesture recognition system, which can recognize a large number of classes with high accuracy and in real time. The main contributions of the thesis on gesture recognition are listed as below:

First, we propose a skeleton-based gesture recognition system which extracts and classifies the features on the frame level, and obtains the final classification result based on the sequence-level modeling in Publication IV. Different from quantizing or learning the code words from features, the direct use of features provides the maximum amount of available information to the system. Together with the effectiveness of the skeletal features, the system provides extremely high accuracies over a large number of classes. Comprehensive experiments have been conducted for several datasets. In Publication VI the system can achieve above 92% in accuracy on a mocap dataset with 65 classes, and for a couple of datasets with a dozen of classes, the system can reach 99% in accuracy. This framework is not restricted to only skeleton data. Due to the frame-level classification, the system minimizes the efforts for the preparation of features, which makes it easily generalize on other data modalities. In Publications VII and VIII the system is expanded for multimodal gesture recognition, that is the RGB, depth and skeleton data. The system is even applied on accelerometer data.

Second, in Publication III we propose simple but effective features for the skeletons from RGB-D and mocap data. A great number of skeletal features has been developed, but many of them require complicated calculation. However, we believe the minimum manipulation of the data

can preserve maximum amount of information given the removal of irrelevant and confusing information. The original 3D coordinates of the skeletons contain all the information of the motion, therefore we still use these joint coordinates but transformed and normalized them in a common coordinates system. It is simple to calculate and computationally light, but keeps all relational information among the joints. We also introduce the idea of preserving the temporal information in the gesture sequence on the frame-level features. It compensates the classification system which does not concern about the temporal information. For multimodal gesture recognition involving hand gestures, we study multiple appearance-based features on low-resolution small hand images with and without hand masks in varying lighting conditions. The results provide useful guidance for selecting the features in similar circumstances.

Third, we use Extreme Learning Machine (ELM) in our classification system, which is a key factor for the success in this accurate and real-time gesture recognition system. In many systems, SVM is widely used for its powerfulness in classification. However, the training is heavy computationally, especially when searching for the most suitable parameters is needed. It consumes both memory and time significantly. The testing is also relatively slow. The nature of SVM makes the system hardly able to perform in real time. Some other classification methods, such as linear regression, is fast in training and testing, however, is fairly low in accuracy. Thanks to the characteristics of ELM, it is extremely fast to train and test with high classification accuracies, which makes the recognition available in real time. For example, the training of ELM on skeletal features for 6000 gestures is about 1 minute and the testing time for each frame is 0.1 milliseconds on a desktop computer. In Publications IV and VI comprehensive comparisons between ELM and other classifiers are studied for the system.

Fourth, we study the relation between skeletons from mocap and RGB-D data. As RGB-D devices are getting more and more widely applied, the skeleton data, traditionally obtained from motion capture, has new resourses from the RGB-D data. It is beneficial to understand the differences between different data resources. In Publication III, a comparison in recognition accuracy was made between skeletons from mocap and RGB-D data for the same actions. Except the difference of the data sources, all the settings are the same in the system. The recognition accuracies indirectly compare these two modalities. Besides the indirect comparison,

Publication V proposed several methods to evaluate the similarity between the skeletons, and the alignment between gestures sequences. As many algorithms have been proposed to extract skeletal models from RGB-D data, the evaluation of the algorithms can be realized by comparison of similarity of the skeletal model with the groundtruth skeletons, where generally mocap skeleton can be considered as the most precise model.

Recognition always corresponds with retrieval. The methodologies used in both tasks are often shared and inspire each other, from the extracted features to the used classifiers. In this thesis, we also build an image retrieval system using local descriptors extracted from images and approximate nearest neighbors (ANN) algorithms. Publications I and II demostrate a mobile augmented reality system which is essentially based on image retrieval. In the proposed system, the users capture photos from magazines with a mobile phone camera application, which sends the images to a server containing a database with multiple issues of supported magazines. Each page of the magazines is stored as a reference image in the database. By searching through the database, the most similar image (page) to the query image is found, and the extra information attached to the page is sent back to the user's application. Different from normal image retrieval systems, the magazine database is highly dynamic. New issues are constantly added and old ones are deleted from the database. Moreover, the images in the database contain mostly text, which generates a large amount of features and increase the memory cost. To solve these problems, we use multiple forests of kd-trees to build a dynamic database, and use descriptors pruning, which significantly reduces the memory cost without sacrificing retrieval system accuracy.

1.3 Structure of the thesis

The thesis is structured as follows. Chapter 2 introduces the research scope on gesture recognition based on the different data modalities, some related topics, and a general description of the image retrieval system developed in Publications I and II. In Chapter 3, we describe the features and classifiers adopted in various gesture recognition systems in the literature. Chapter 4 gives a detailed description of our proposed gesture recognition system for skeletal data, which is developed in Publications III and IV. The visualization of the skeletal features developed in Publication VI and the alignment between mocap and Kinect skeletal data from Publication V

are also described in this chapter. Publications VII and VIII study gesture recognition using multimodal data from a Kinect device, which is described in Chapter 5. Chapter 6 then summarizes the thesis.

2. Action recognition

Action recognition is an active and challenging research topic. In this chapter, we will introduce different data modalities used in the recognition task and some related topics to action recognition.

2.1 Data modalities

For the last two decades, there has been immense research conducted on the task of action recognition but mainly dealing with RGB video data [101]. Many public datasets available online [44, 133, 130] provide a platform for researchers to evaluate their methods on common benchmarks. In these datasets, a few different natural actions are usually to be recognized, such as “walking”, “running” and “hand waving”. Each instance of an action is recorded into a separate video file containing multiple frames. Figure 2.1 shows some examples from the KTH action database [133]. Each image is one frame from the video file, showing the general contents of the video.

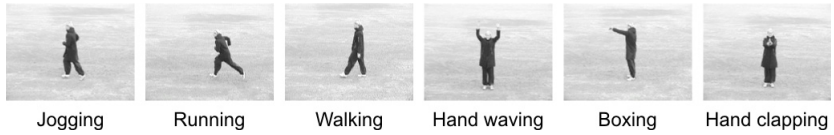


Figure 2.1. Some examples of sequences corresponding to different types of actions in the KTH action database.

A large number of methods has been proposed to solve this research question. One direction is to extract skeleton models from the video data and recognize the actions based on the skeleton data [93, 144]. In this thesis, we do not work on the recognition of actions from RGB video data, but from two other data modalities: motion capture data and RGB-D data. Furthermore, we use the same framework to experiment with accelerometer data to recognize actions.

2.1.1 Motion capture data

A motion capture system consists of multiple calibrated high-resolution cameras set up in a dedicated space. The actor wears multiple markers on his body and performs actions within a certain area. Multiple videos are recorded simultaneously in multiple angles. The data is processed by specific software sold together with the system, and highly precise modeling of the main joints of human skeletons is achieved. The hardware and software are very expensive which make motion capture data very costly. Motion capture is often used in fields such as filmmaking, computer animation, biomechanics, gesture analysis, game development and sports science. Figure 2.2 shows a recording process by an OptiTrack motion capture system and the skeleton model generated by the proprietary software.

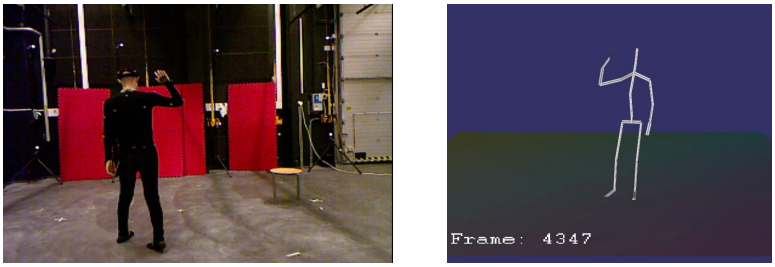


Figure 2.2. The OptiTrack motion capture system (left) and a generated skeleton model (right).

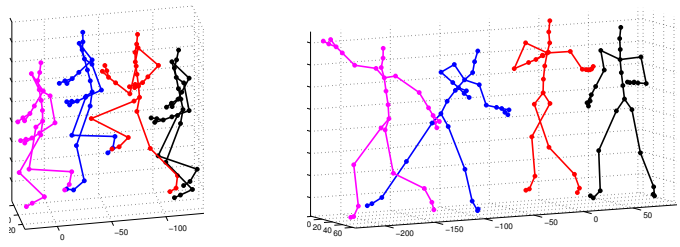
Different systems often generate data on different formats and use different skeleton models. Commonly used formats include ASF/AMC, BVH and C3D [96]. The mocap data provides detailed motion information, for example the translation and rotation information related with the joints, but it does not provide directly the joint 3D coordinates. Different data formats require different methods to calculate the joint coordinates [96]. The number of joints in a skeleton model varies on different mocap systems. For example, the skeleton model from OptiTrack has 20 joints.

Motion capture databases

Fortunately even though the mocap data is very expensive to obtain, there are several large motion capture databases available online for research. The Carnegie Mellon University Motion Capture Database (CMU database) [26] is one of the largest mocap databases. It contains comprehensive amounts of motion data, covering physical activities, sports, human interaction and interaction with environment, and so on. However, due to the large size of the database and the large number of actions in the database,

researchers often use a partial database to conduct their experiments. Because the data in the dataset is not well marked, this makes it difficult for other researchers to use exactly the same data to compare the different methods.

The Motion Capture Database HDM05 [107] is another popular database. It contains actions in five categories. The recordings are manually cut out and arranged into 130 action classes. The names of these actions can be found in the Appendix of Publication V. Most of these classes contain 10 to 50 different instances amounting to roughly 1500 motion sequences and 50 minutes of recording. The whole database is clearly labeled and organized. Figure 2.3 shows a couple of actions from the HDM05 database. The skeleton model has 31 joints.



(a) **(b)**

Figure 2.3. Example actions from the HDM05 dataset. (a) Hop left leg; (b) throw far.

The TUM Kitchen Data Set [147] intends to provide realistic natural motions for recognition and understanding of natural everday human activities. It records setting up a table performed by several actors in different ways in a kitchen environment. It requires the segmentation of the sequences into semantic classes and then recognition of the actions. The HumanEva-I dataset [136] contains both video and mocap data. It contains six common actions: walking, jogging, gesturing, throwing and catching a ball, boxing and combo (a series of actions) by four actors.

Among these databases, HDM05 contains a large variety of actions and is well organized, and it is also used by several researchers (e.g.[152, 105]). Therefore, in Chapter 4 HDM05 dataset is used to evaluate our recognition system, and the results are compared with repeated results from different systems with the same data.

Table 2.1 summarizes the above mentioned motion capture databases with descriptions of the characteristics of the actions and challenges from the aspect of recognition.

Table 2.1. Summary of motion capture databases. NC: Number of classes.

Database	NC	Actions	Challenge
CMU database	> 100	A huge number of actions, including locomotion, activities, sports, human interaction, actions in multiple scenarios and environments.	Actions are not well organized, and some actions only have a few samples.
HDM05	130	Commonly performed full body actions of a single person, such as kicking, sitting down etc.	Some actions are similar to each other.
TUM Kitchen Data Set	10	A series of natural movements in a kitchen, such as taking something, opening/closing a door/drawer, carrying etc.	The data consists a series of actions, which requires the segmentation of the actions first.
HumanEva-I dataset	6	Walking, jogging, gesturing, throwing and catching a ball, boxing and combo.	-

2.1.2 RGB-D data

In the last few years, the commercial RGB-D sensors such as Microsoft Kinect and Asus Xtion have prevailed among researchers and normal consumers due to their low cost and high functionality. The RGB-D sensor, providing both RGB video and 3D depth data in a compact device, has raised a technical revolution in many classic problems of computer vision research. Prior to the Kinect, to capture accurate 3D depth data of a scene, a 3D laser scanner was the main device to be applied in non-contact measuring situations. However, the massive volume and price of the laser scanner limit the potential usage in many applications. Instead a stereo vision system consisting of two cameras is often employed to get 3D information, for example in robotics [75, 71]. Nevertheless, the resolution of the cameras, the calibration of the system and the required heavy computations increase the complexity of the system and significantly influence the accuracy of the 3D depth data.

The Kinect device was initially released for the Microsoft Xbox 360 console gaming system. It can recognize the whole body and build an avatar of the player, so the player can play full body games without any controller. Nowadays, Kinect has been widely used to replace RGB cameras to provide new opportunities in many applications. For example, by providing depth information, image segmentation with the RGB-D data has more options for solutions [127]. The RGB-D sensor enables more flexibility [173] for sign language analysis which previously has often been conducted in prepared environments. It has been used, e.g., in a system to support communication between deaf and people with normal hearing by recognizing

the sign language [20]. Robots can be controlled by hand or body gestures through gestural recognition [121, 125]. Similarly, users can interact with a computer system based on gestures without touch-based or other input devices [3].

Kinect device

Figure 2.4 shows the physical appearance and sensor components of the first generation of the Kinect device. The color sensor is a RGB camera. An infrared (IR) emitter emits infrared light beams, and the reflected IR beams from the environment come back to the IR depth sensor. The distances between different objects and the sensor are obtained based on the reflected beams. A multi-array microphone containing four microphones can be used for capturing sound. The tilt motor is capable of vertically tilting the sensor bar with a range of $\pm 27°$.

The video streams use the VGA resolution (640×480 pixels) with 8 bits per channel for the RGB video and 11 bits for the depth video. The maximum frames per second (fps) can reach 30 fps. The angular field of view of the sensor is $43°$ vertically and $57°$ horizontally. The optimal sensing distance ranges from 1.2 meters to 3.5 meters.

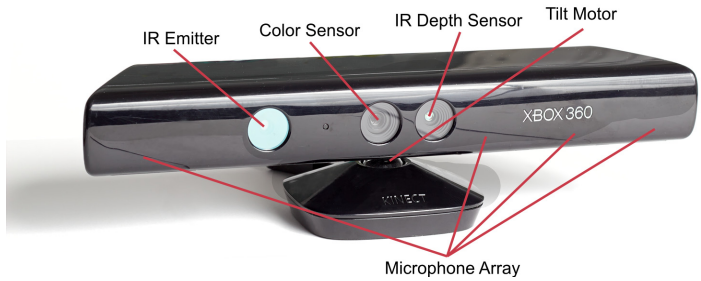


Figure 2.4. The Kinect device.

Skeleton modeling from Kinect

In addition to the depth information provided by Kinect, another data modality is also often provided – the skeleton model. Comparing to extracting the skeleton model from RGB videos, the depth information makes the extraction more feasible and stable. Several algorithms have been proposed and applied to extract the skeleton from the depth data [43, 135, 143]. The basic idea underlying these methods is to segment the human depth image into multiple body parts with dense probabilistic labeling. This segmentation of the body parts can be considered as a classification task for each

pixel in the depth image. The 3D joint positions are computed based on the spatial modes of the inferred per-pixel distribution.

Due to the characteristics of the algorithms, the skeletal tracking is optimized for when the user is facing the Kinect. Sideways poses with parts of the user invisible to the Kinect make the skeletal tracking challenging. In [87], the performance of the skeleton from the Microsoft SDK is measured from the aspects of noise, accuracy, resolution and so on. However, the ground truth for the measurements is not from the true skeletal model, but instead from some measurements with physical measuring tools, e.g. a wooden meter stick.

The skeleton generated from the depth data is less accurate and stable compared to mocap data. Still, one can adopt the methodology developed for mocap skeletons to work with RGB-D skeleton data as well in most cases. Different algorithms often generate different skeleton models. For example, the Kinect for Windows SDK [158] provides skeletons with 20 joints, and the NiTE library [118] generates skeletons with 15 joints.

RGB-D Databases

There are many public benchmarking databases containing depth information available online, which provides a good platform for researchers to concentrate on the recognition of actions rather than on the data collection. Some of the databases only provide the RGB and depth videos, while some also provide the skeleton data.

Table 2.2 shows a summary of existing databases with RGB-D data, with the exception of the MSR Action3D Dataset which only contains the depth video. These datasets emphasize different application purposes, and therefore the action categories vary from each other. The Hollywood 3D dataset [50] focuses on action recognition in natural environments. It collects actions from feature films mostly in unconstrained situations. RGBD-HuDaAct [109] concentrates on the daily activities of senior citizens for assisted living in health-care. ACT4^2 [24] focuses on daily living activities with clear semantics in real life. MSRGesture3D [74] contains 12 dynamic American Sign Language (ASL) gestures. The MSR Action3D dataset [83] collects 20 actions in the context of interaction with game consoles, with various movements of arms, legs, torso and their combinations covered in the actions.

Table 2.2. Databases of RGB and depth video. NP: Number of performers; NC: Number of classes.

Database	NP	NC	Actions
Hollywood 3D dataset [50]	-	14	run, punch, kick, shoot, eat, drive, usephone, kiss, hug, standup, sitdown, swim, dance, NoAction
RGBD-HuDaAct [109]	30	12	make a phone call, mop the floor, enter the room, exit the room, go to bed, get up, eat meal, drink water, sit down, stand up, take off the jacket, put on the jacket
ACT4^2 [24]	24	14	Collapse, Drink, MakePhonecall, MopFloor, PickUp, PutOn, ReadBook, SitDown, SitUp, Stumble, TakeOff, ThrowAway, TwistOpen, WipeClean
MSRGesture3D [74]	10	12	Bathroom, Blue, Finish, Green, Hungry, Milk, Past, Pig, Store, Where, Letter J, Letter Z
MSR Action3D Dataset [83]	7	20	high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, side-boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pickup and throw

RGB-D Databases with skeleton modality

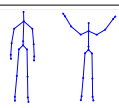
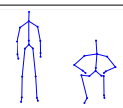
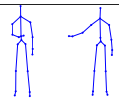
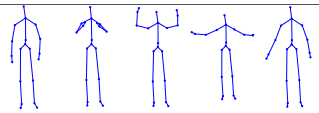
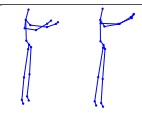
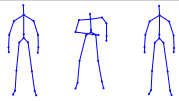
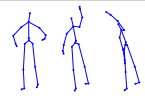
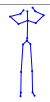
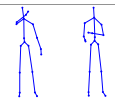
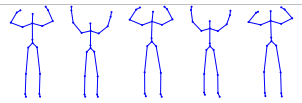
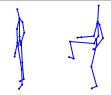
In addition to the depth data, some datasets also provide skeleton data. The MSR Daily Activity 3D dataset [156] is designed to cover daily activities of humans in the living room. It contains 16 activities such as drink, eat, and read book, and most of the activities involve human-object interactions.

The Microsoft Research Cambridge-12 (MSRC-12) Kinect gesture data set [39] is collected for the evaluation of the effects of different instructions to performers in gestural training systems. In the dataset, five different kinds of instructions are given to a total of 30 performers for conducting the same kind of actions. The description of the actions can be seen in Table 2.3. These actions can be divided into two categories: actions for the control of software in a HCI environment, and actions for surveillance purposes. The dataset is used in Publication IV for the validation of our system.

2.1.3 Accelerometer data

Accelerometer data is another modality gaining more attention for action recognition. An accelerometer can measure acceleration forces, either static, like the constant force of gravity, or dynamic, caused by motion or vibration. It is widely embedded in smartphones, tablets, or built in wearable motion sensor networks [138, 165]. Due to the easy access to the accelerometer data from these smart devices, using the data for

Table 2.3. The MSRC-12 Kinect gesture data set and example motions. From Publication IV.

Metaphoric gestures	Main frames	Iconic gestures	Main frames
Start music\ raise volume (G1)		Crouch or hide(G2)	
Navigate to next menu(G3)		Put on night vision goggles(G4)	
Wind up the music(G5)		Shoot with a pistol(G6)	
Take a bow to end the session(G7)		Throw an object such as a grenade(G8)	
Protest the music(G9)		Change weapon(G10)	
Lay down the tempo of a song(G11)		Kick to attack an enemy(G12)	

the recognition of human activities has wide application fields, especially for remote health care [68]. For example, by using the accelerometer to monitor daily physical activities and the built-in camera to analyze food intake, an application can remind the user in real time about required daily activity to balance the energy taken to keep healthy [42]. In [141], two tri-axial accelerometers are mounted around wrists to distinguish the type of skin scratching either as caused by a medical condition or as a routine response, such as to an insect bite or uncomfortable clothes. But, on the other hand, the acceleration data obtained from a single device is usually not very distinctive, and therefore the number of activities that can be accurately recognized is typically very small [81, 49].

The Berkeley Multimodal Human Action Database (MHAD) [112] is a multimodal database which contains data from a motion capture system, a camera system, a Kinect system, an audio system and accelerometers. Six accelerometers are attached on the actors' wrists, ankles and hips. The Kinect data includes the video streams without the skeleton data. The database contains 11 classes of actions performed by 12 actors, totalling

about 660 instances. The actions are performed with the full body, such as jumping, bending, punching, throwing and waving. This dataset provides a better collection of acceleration data to recognize a relative big variety of actions. We use both the mocap data and the accelerometer data to evaluate the performance of our system compared with other published methods.

In Chapter 4, we introduce our action recognition system based on skeletal data, both from mocap and Kinect. More detailed information can be found in Publications III, VI and IV.

2.2 Hand gesture recognition

The action recognition discussed above mostly involves whole body movements. In many cases, however, the distinction between actions only relies on the movement of arms and hands, which is commonly referred to as hand gesture recognition. It is largely used for HCI and sign language recognition. The recognition of hand gestures can also be grouped into static hand gestures or postures and dynamic hand gesture recognition. Previously the research on this field has mostly focused on static hand gesture recognition [40, 23]. Figure 2.5 illustrates some defined static hand gestures used in [37].

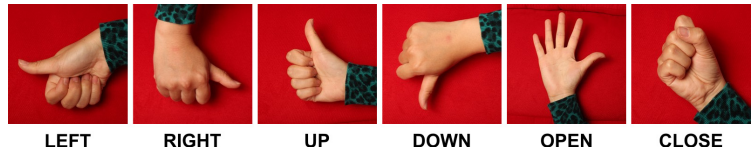


Figure 2.5. Some static hand gesture definitions.

A dynamic gesture is composed of multiple static gestures. The series of movements represents a complete gesture, as seen in Figure 2.6. Compared to the static gestures, intuitively the motion of the hands and arms provides further options to be used as features for the recognition. The commonly used motion features include trajectory [69, 167], location, orientation [33] and velocity [171].

In addition to the motion-based features, most research focuses on the recognition of hand postures, that is, extracting features directly from the hands. This hand modeling method can be roughly divided into two categories: 3D hand model based modeling and appearance based modeling [120, 34]. The former tries to build a hand volumetric model or the skeletal

(a) **(b)**

Figure 2.6. Some dynamic hand gesture definitions similar as the ones from [61]. (a) whap-whap; (b) behind.

model, which are often challenging and computationally costly [142, 30]. However, with the aid of the RGB-D sensor, 3D hand modeling becomes more robust and can even be achieved in real time. These methods often label each part of the hand and consequently provide the skeleton data [114, 67, 66]. The depth information improves significantly the segmentation of the hand from the background, and makes even the recognition of the finger tips feasible [84, 123, 128]. The fingers encode detailed information about the hand posture, often leading to building a 2D hand posture skeleton. However, in these cases, the hands are typically streched from the body in order to ease the segmentation of the hands from the background [84].

The appearance-based methods usually have smaller computational costs, and a big variety of features can be extracted from the hand regions [140, 162, 146]. These include color [12], contour, silhouette [172], location, and orientation of the fingertips. The limitation of the appearance-based methods is that they can only recognize a discrete number of hand gestures, which is determined by the training data.

These hand features obtained from the RGB image and the depth information also have some physical constraints for the settings of the applications. To build the 3D hand model or obtain precise hand information, the size of the hand images has to be large enough considering the resolution of the sensors. The hand region sometimes dominates the whole image [8], or at least occupies 100×100 pixels [114, 122]. And when skin color is used for hand segmentation, a good lighting condition is often required [51].

When the hand size gets extremely small, detecting the hand and extracting precise hand features becomes more difficult. In this case, combining the hand features and body movement is essential for gestural recognition. In Chapter 5 we will illustrate our work on this topic.

2.3 Research topics related with action recognition

As a comprehensive research topic, action recognition also relates with other research topics.

Action spotting
In action spotting, or gesture segmentation, the task is to find the start and end of a gesture. It is a primary requirement for gesture recognition in many systems where the beginning and end of the gestures are clearly defined. Some systems are designed for simultaneous segmentation and recognition of the gestures [2]. When gestures are performed continuously, the temporal segmentation of the gestures is crucial for correctly recognizing the gestures. For mocap data, applications such as animation, commercial films and video games all require precise segmentation of the gestures [4]. Skeletal data has the advantage of simple calculation of the motion features, e.g. trajectory, and velocity. Therefore, even in the RGB-D data, the skeletal features are often used for segmentation [108, 170]. When the gestures are segmented by resting positions, the resting position can also be considered as a gesture and therefore can be learnt by the classification system. HMMs are often adopted for this purpose [170, 25]. If the gestures are performed continuously without resting positions, the segmentation becomes more complicated, and probabilistic PCA is often applied [4, 176]. On the other hand, the sliding window is a universal method even though in most cases it is not a perfect solution [7]. Broadly speaking, action recognition can be considered as the combination of action spotting and action classification. However, action classification is also often called action recognition, while action spotting is explicitly studied as an independent research topic. In this thesis, action recognition is equivalent to action classification, and we refer to action spotting explicitly.

One-shot-learning
Humans can usually easily recognize a gesture or an object after just seeing an example once. This can be called one-shot-learning, and it has been a popular topic in object recognition [77, 38]. One-shot-learning provides the possibility of consumer applications of gesture recognition, where the consumers need only to perform one example of each gesture. However, for most gesture recognition systems, a sufficient amount of training data is required for accurate training of the system. In the ChaLearn gesture challenge 2011/2012, the task was to recognize gestures with only one

example of each class as the training data [46]. Each task contained a small vocabulary of 8 to 12 arm and hand gestures with only the RGB video and depth data recorded by a Kinect device. Dynamic time warping, nearest neighbor [160, 154, 174], and correlation coefficient [92] are commonly selections in this situation [48]. [90] characterizes an action as a point on a product manifold and employs geodesic distance for the classification. Though the support vector machine (SVM) usually requires a large amount of training data, by extracting multiple instances of features from one training sample, SVM performs impressively well with such a small set of training data [36]. Hidden Markov Models, Conditional Random Fields, and other similar graphical models are also applied for one-shot-learning to perform the gesture segmentation and recognition at the same time [46]. In this thesis work, the datasets we tested have sufficient training data, and therefore one-shot-learning is not considered.

Action retrieval

Another research topic closely related to action recognition is action retrieval, also denoted as motion retrieval for motion capture data. Because of the dearly cost of mocap data, it is desirable to be able to reuse data from pre-recorded databases [85]. Action recognition can also be directly applied for retrieving similar motions [106, 5]. In addition, the skeletal features used for gesture recognition and some variant of DTW can also be used in retrieval systems [72, 65, 73]. In this thesis, we do not apply our gesture recognition methods directly for motion retrieval. However, in Publication V we use subsequence DTW to align the motion sequences between Kinect skeletal data and mocap data, which provides the possibility to retrieve mocap data by using Kinect skeletons as queries.

a) b)

Figure 2.7. An example matching: (a) the query photo, (b) the matching magazine page. From Publication I ©IEEE.

Action retrieval from video is a relatively new topic [63, 62]. Because of the characteristics of video and image data, some features used in image retrieval can also be adopted in action retrieval systems [17]. Nowadays smart phones are ubiquitous and many interesting research prototypes and useful applications of image retrieval systems are developed for mobile phones [148, 22, 21]. Publications I and II present an image retrieval system for a mobile phone application. In this system, a query image taken from a commercial magazine with a mobile phone camera is sent to the server. By searching through the image database in the server, the best matching image is determined, and the extra information attached to it is sent back to the user's mobile phone, so that the user can get more information of the object that she is interested in. Figure 2.7 shows an example of matching between a query image taken by a mobile phone camera and a magazine page in the server database. The methodologies of image retrieval, such as the features and matching, can also be used in the gesture recognition. Therefore, in addition to describing the gesture recognition system in this thesis, we also briefly introduce the methods in Chapter 3 used in the image retrieval system described in Publications I and II.

3. Overview of action recognition systems

This chapter presents a brief review of existing approaches proposed for action recognition systems. The structure of a recognition system is usually determined by the characteristics of the used features and classifiers. Therefore, in this chapter we review features and classifiers separately, while the whole system is revealed indirectly through the introduction of both features and classifiers.

3.1 Feature extraction

In Section 2.1, we have introduced the data modalities commonly used in a gesture recognition system. In this section, we give an overview of the features extracted from these data modalities categorized into skeletal features, image features and depth features.

3.1.1 Skeletal features

The skeletal features can be extracted both from the mocap data and from the skeletons generated from RGB-D sensor data. The skeletal data includes the 3D coordinates of all skeleton joints and sometimes also the angular information associated with the joints.

The features extracted from the 3D coordinates are the most frequently used features from skeletal models. [134, 157] directly use the x, y, z coordinates of the joints without any post processing, but since the coordinates are related to many external factors other than only the gestures, these systems can usually only recognize a small vocabulary of actions in very restricted environments. Therefore, a large variety of normalization methods have been applied to the original 3D coordinates in order to be invariant to different human performers and the coordinate system used in the recording. For mocap data, the position and orientation of the root joint can be set

to zero to normalize the other joints [73, 91]. [129] uses the neck joint as the origin of the coordinates and updates the coordinates of the remaining joints based on the new origin. [19] normalizes the coordinates with the distance between the left and right shoulders, followed by subtracting the shoulder center from the other joints. [175] transforms the root to the origin and aligns the vertical axis of the human body to be parallel to the y axis.

In addition to the normalization of the original coordinates, calculating the distances between each pair of joints is another popular method [152, 156]. It automatically eliminates the influence of the coordinate system, leaving only the size of the performer to be normalized. [76] calculates the distances between the hands, elbows, and the spine in each dimension separately with the time index of the frame to form a feature vector. In [168, 177], the pairwise distances between joints in the current frame, the pairwise distances between joints in neighboring two frames, and the pairwise distances between joints in the current frame and the first frame are concatenated into one feature vector for each frame. Furthermore, [106] describes the geometric relations between certain joints as a binary value, forming a 39-dimensional boolean vector for each frame. The whole action is then represented by a feature matrix, denoted as motion template. In addition to the distances between the joints, motion features, such as velocities, and trajectories, can also be easily calculated from the joint coordinates [125, 52].

Compared to the 3D joint coordinates, the angular data represents the relative information between the joints. It is independent from the size of the actors and the coordinate system of the recording device, and therefore does not require normalization. [32] combines the 3D angles of 20 joints from a single frame to form one feature vector for that frame. The feature vectors from each frame in the action form a feature matrix to represent the action. In addition to directly using the joint angles, there is also a big variety in calculating the angular information between the joints. [112] calculates the joint angles from 21 joints in the skeletal hierarchy, which forms a 21-dimensional feature vector for each frame. [166] builds a coordinate system at the hip center joint, and calculates the 3D angles of the remaining joints in the new coordinates. These 3D angles are then combined together to form the feature vector of each frame. Similarly, [127] builds reference coordinates based on the torso joint. Eight effective joints are selected and each of them is represented by two-dimensional

angular coordinates. Together with the 3D angles of the torso, these angles form a 19-dimensional feature vector for one frame. [164] sets a spherical coordinate system around the skeleton and partitions the spherical volume into bins. The joints are projected into these bins with probabilistic voting, and these votes are accumulated over the joints. As a result, each frame is represented as a histogram.

Some of these features are of high dimensionality, and therefore they are often post processed by PCA to reduce the dimensionality [152, 168]. We have instead used an autoencoder [54] to visualize our features on a 2D plane. Detailed information of the proposed method can be found in Publication VI.

3.1.2 Image features

As discussed in Section 2.2, for hand gesture recognition we focus on appearance-based hand features. A large number of features can be extracted from RGB or grey-scale images. These features are mostly computed for local interest regions. For hand gesture recognition, the regions of interest are naturally the hand regions. However, in general applications these interest regions can be obtained by a series of mathematical calculations, which are often referred to as interest point detection. The commonly used methods include Harris corner detection [53], Hessian-Affine regions [98], maximally stable extremal regions (MSER) [94], Laplacian of Gaussian (LoG) [16] and Difference of Gaussians (DoG) [89]. After the detection of the interest points, various feature descriptors can be extracted from the regions around the interest points. These descriptors should be highly distinctive and repeatable and invariant to illumination, 3D viewpoints, etc.

Among the many proposed methods, the scale-invariant feature transform (SIFT) [89] is the most commonly used one and an effective method to transform image data into scale-invariant local features [99]. It detects the local maxima and minima of the DoG images as keypoints candidates, followed by detailed models fitted to determine the location and scale to derive the final keypoints. One or more orientations are assigned to each keypoint, so the keypoint descriptor can be extracted related to the orientation to achieve rotation invariance. In order to calculate the descriptor for the keypoints, a window of 16×16 pixels is approximately centered on the keypoint, and a 2D Gaussian function with standard deviation of one half of the width of the descriptor window is used as a weighting function.

The gradient magnitude and orientation in the window at each pixel are calculated. The descriptor window is divided into a 4×4 grid with the size of each cell as 4×4 pixels. The samples in each cell are accumulated into orientation histograms with 8 bins. The dimensionality of the feature vector is thus 4×4×8=128.

Another popular scale- and rotation-invariant detector and descriptor is SURF (Speeded-Up Robust Features) [6]. By operating on integral images and simplifying the methods for the detection and descriptor extraction, SURF can be computed much faster than SIFT with approximately equal performance. Both methods are widely used in object detection [88], panorama stitching [14], 3D scene recognition [13], etc. In Publications I and II, we have used SIFT in our prototype design and SURF in the final released version of the content-based image retrieval system.

Apperance-based hand features

The approaches to hand gesture recognition can roughly be grouped into appearance-based and model-based methods. The latter usually requires good lighting conditions and relatively high resolution images. In this thesis, the applications of the hand gesture recognition have relatively poor lighting conditions and very low image resolution. Therefore, we only focus on the appearance-based hand features.

The SIFT and SURF features are extracted based on interest points with different scales and orientations. For the hand feature extraction, however, the hand regions are detected beforehand and the problem is to select a suitable descriptor to transform the hand image into a feature vector. In Publication VIII, we tested several features of the hand images in our recognition system. These features are introduced below, and a detailed performance evaluation of them can be found in Publication VIII.

Histogram of Oriented Gradients (HOG) [28] was originally proposed for human detection. After its successful application on pedestrian images, HOG features have been widely applied on many kinds of images. There has been some previous work on applying HOG on hand images [139, 117]. Similarly as SIFT, HOG also extracts orientation histograms. SIFT features are typically extracted around interest points and the whole image is commonly represented as a bag of local features. On the other hand, HOG, as a dense descriptor is computed across the whole image. Figure 3.1 shows an example of the computation of the HOG features. In this example, the image size is 168×168 pixels. It is divided into a 21×21 grid

with the size of each cell as 8×8 pixels. The gradient vector for each pixel in the cell is calculated and the orientations of the gradient vectors are stored into a 9-bin histogram ranging from 0 to 180 degrees. According to [28], the magnitude of the gradient vector is added to the histogram bins. After calculating the histograms for all cells, the cells are grouped into blocks of size 2×2 cells, and the histograms are concatenated into one feature vector. By L_2 normalization of the histogram, the feature becomes invariant to illumination changes. The blocks are formed by sliding a window of size 2×2 cells across the whole image and maintaining a 50% overlapping with each neighboring block. The final HOG descriptor is then obtained as the concatenation of all block histograms.

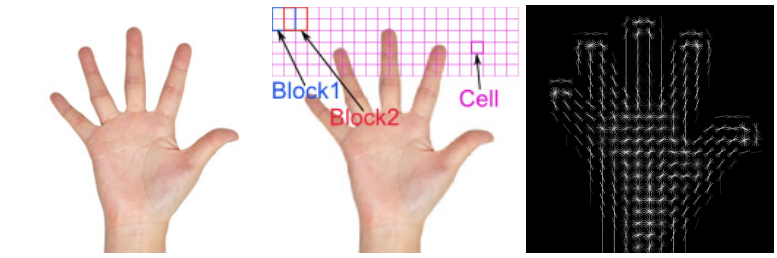
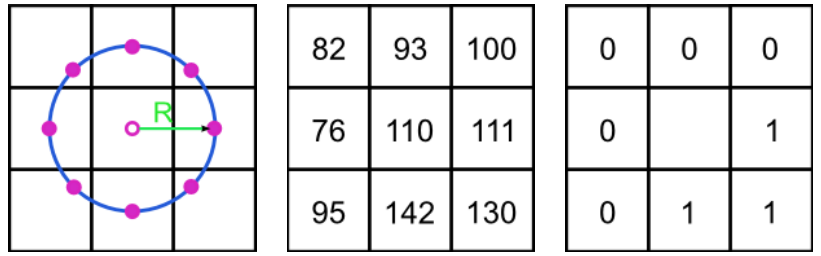


Figure 3.1. An example of HOG feature extraction. (a) A hand image; (b) some cells and blocks assigned to the hand image; (c) visualization of the HOG features.

Another effective descriptor, especially for texture analysis, is local binary patterns (LBP) [115, 116]. It is theoretically and computationally simple but robust in terms of grey-scale variations, and it has successfully been applied e.g. in texture classification, face recognition [56] and action recognition [64]. The basic form of LBP can be calculated as shown in Figure 3.2 with two parameters R and P required to be decided. R is the radius of a circle centered in the pixel of interest, and P is the number of pixels equally spaced on the circle. If the coordinates of the center pixel are $(0, 0)$, the coordinates of the P neighboring pixels are given by $(-Rsin(2\pi p/P), Rcos(2\pi p/P))$, where $p = (0, 1, ..., P-1)$. If the coordinates do not fall exactly in the center of the pixel, the gray values of these neighboring pixels are calculated by interpolation. The intensities of these P neighboring pixels are compared to the center one: if the center pixel's value is smaller than the neighboring pixel value, then the value 0 is assigned on the position of that pixel in the feature; otherwise the value 1 is assigned. Figure 3.2 shows an example with $R = 1$ and $P = 8$. In

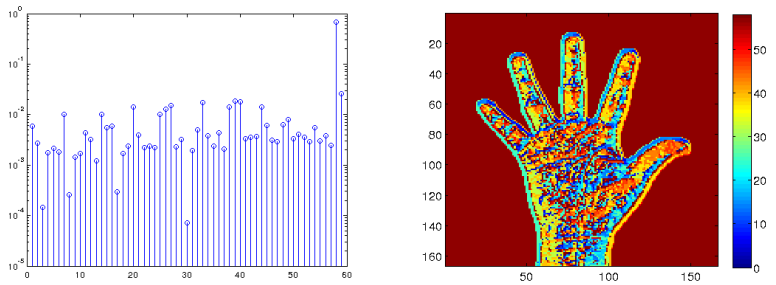


(a) **(b)** **(c)**

Figure 3.2. An example of LBP extraction. (a) The center and its neighboring pixels with $R = 1$ and $P = 8$; (b) the grey values of a 3×3 image; (c) the signs of difference between the neighboring pixels and center one, 0 if it is negative, otherwise 1.

this example, LBP forms an 8-bit binary digit, and a total of 256 different patters can be generated.

Furthermore, an uniformity measure U can be introduced to detect the "uniform" patterns, which corresponds to the number of transitions, that is, bitwise $0/1$ changes between the successive bits in the circular binary patterns. For example, patterns 00000000_2 and 11111111_2 have $U = 0$, while the pattern in Figure 3.2(c) has $U = 2$. By definition, the uniform patterns should have $U \leq 2$. Thus, patters with $U > 2$ are all grouped into one group whereas the rest are indexed as before. Therefore, the 8-bit LBP of 256 patterns is decreased to 59 patterns. The final feature of the image is the histogram of the LBP features generated on each pixel in the image. Figure 3.3(a) shows the normalized histogram of LBP features extracted from the hand image in Figure 3.1(a), and Figure 3.3(b) visualizes the LBP features of each pixel in the image. Similarly to HOG, we can also divide the hand image into cells, calculate the LBP histogram for each cell, and concatenate the histograms to form the feature vector for the hand image.



(a) **(b)**

Figure 3.3. A LBP example of a hand image. (a) The normalized histogram of uniform LBP for the hand image; (b) a visualization of the LBP pattern for each pixel.

Gabor filter responses are successfully used in many computer vision tasks, such as edge detection and texture representation. The 2D Gabor filter [29] can be represented as

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}) \exp(i(2\pi \frac{x'}{\lambda} + \psi)), \qquad (3.1)$$

where $x' = x\cos\theta + y\sin\theta$ and $y' = -x\sin\theta + y\cos\theta$. In this equation, λ represents the wavelength of the sinusoidal factor, θ specifies the orientation of the normal to the parallel stripes of the Gabor function, ψ is the phase offset, σ is the standard deviation of the Gaussian factor and γ is the spatial aspect ratio which specifies the ellipticity of the Gaussian factor. In feature extraction, Gabor filter banks are often built as multi-resolution structures consisting of a set of Gabor filters tuned to different frequencies and orientations. An example of the hand image in Figure 3.1(a) convolved with a bank of Gabor filters is shown in Figure 3.4. In this example, four scales in frequency and in orientation are used, which is adopted from [122].

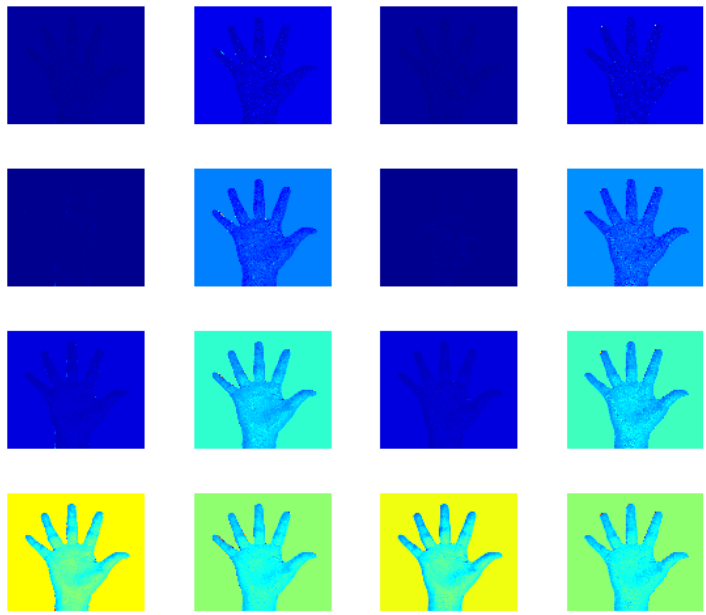


Figure 3.4. A hand image convolved with a bank of Gabor filters.

From Figure 3.4 we can see that a Gabor filter bank can successfully capture the edges and corners of the hand image. In order to convert the convolved images into a feature vector, one method is to directly transform the convolved image matrix to a feature vector. However, due to the size of the image, the feature vector could be of very high dimensionality.

In some cases, the convolved matrices are first downsampled and then transformed into vectors. [122] uses another method to obtain the feature vector. The convolved images are divided into grids, and a bank of 2D Gaussian functions is built with the mean vectors located at the center of each cell and with the same standard deviation. Figure 3.5 illustrates a bank of Gaussian functions located in a 2×2 grid matrix. The matrix is obtained by an element-by-element multiplication of the Gabor filtered image and the Gaussian function, and the sum of all elements in the matrix is returned. The same operation is applied to all Gabor filtered images and the banks of Gaussian functions, and the values are concatenated to form the final feature vector for the image.

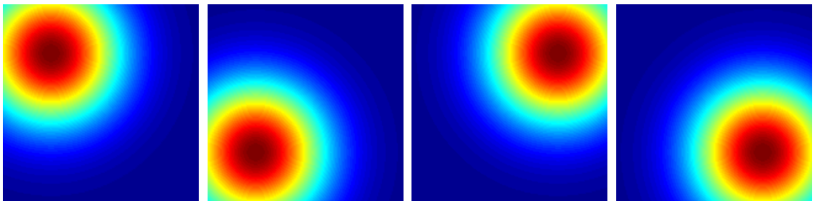


Figure 3.5. A bank of Gaussian functions with mean vectors located in a 2×2 grid matrix.

In addition to the above image features, we also use the histogram of oriented 3D spatio-temporal gradients (HOG3D) [70] in Publication VIII. HOG3D is often used for action recognition in videos, where the videos are represented by bags of HOG3D features extracted on 3D points sampled in the X and Y (spatial) and T (time) dimensions. The calculation of HOG3D can be roughly described as follows: (1) A cuboid support region is built centered at the interest point, and it is divided into $M \times M \times N$ cells in 3D, where M and N are positive integers. (2) Each cell is further divided into $S \times S \times S$ subblocks. (3) A 3D gradient vector is calculated for each subblock and quantized by a regular polyhedron. (4) A histogram for a cell is obtained as the sum of all quantized gradient vectors of all subblocks. (5) All histograms from each cell are concatenated and normalized to form a HOG3D feature. In our experiments in Section 5.2.4, we build a support region around the hand joint coordinates and extract HOG3D features for each frame of the video.

3.1.3 Depth features

Compared to RGB color images, depth data captures the geometric and shape information, which can be more distinctive than the color images.

Unlike the color image, it is not affected by illumination, which removes the requirement of good lighting conditions. Some research has been conducted on using pure depth data for action recognition. The depth data can be considered as 2D depth image or as 3D cloud points representing the depth information. In image form, the recognition methods for image data can often be directly applied to the depth data. As 3D cloud points, many new features have been invented based on it. Therefore, there has been a variety of features and recognition methods developed for depth data.

Silhouettes, expressing the shape of postures, are an effective input for video action recognition. [83] extends this concept to depth data and uses an action graph to model the human actions. The 3D depth points are considered as a silhouette for each frame. The posture is modeled as the joint distribution of these points. Due to the huge number of points in each frame, the computation is extremely heavy. Therefore, a subset of 3D points are selected based on three orthogonal Cartesian planes for each frame. The results show a significant improvement compared to a 2D silhouette.

Another widely used approach is to extract features from the spatio-temporal interesting points (STIP) [78] and represent the actions using Bag of Visual Words (BoVW) [111]. [24] adopts this approach but uses a new depth feature named comparative coding descriptors (CCD), which is inspired by the LBP descriptor. Taking the interest point as the reference point at the center of a cuboid of size $3 \times 3 \times 3$, subtracting the depth value of the surrounding points from the reference point, and comparing with a threshold, the feature encodes a 26-dimensional vector with elements of $[-1\ 0\ 1]$ only. Compared to HOG and HOF (Histogram of Optical Flow) [79] features extracted from the RGB video, and HOG-HOF features from the depth video with the same STIPs, the accuracy with the CCD feature outperforms the other schemes for the ACT4[2] [79] dataset.

While STIP was originally developed for RGB video, [163] proposes an algorithm for extracting STIPs from depth videos (DSTIPs) and a local depth feature named depth cuboid similarily features (DCSF), which are based on self-similarity to encode the spatio-temporal shape of the 3D cuboid. [155] proposes a semi-local depth feature called random occupancy pattern (ROP), which considers a depth sequence as a four-dimensional volume (x, y, z, t) and selects subvolumes based on discriminability. In each subvolume, if there is a cloud point in this 4-dimensional space, then

a counter is increased by one for that position. The points within the subvolume are summed, and the result is given as an input to a sigmoid normalization function. The output of the function is the ROP feature. For each gesture, a fixed amount of subvolumes, that is, a fixed number of ROP features, are selected.

Instead of extracting features from parts of the depth 3D point cloud, [169] takes all depth data into use to build depth motion maps (DMMs) and to extract HOG features from three DMMs, concatenated as the final representation for the action. In the depth sequence, each frame is projected onto three orthogonal Cartesian planes to generate three 2D maps. The differences of the 2D maps between neighboring frames are accumulated in each projected plane, and therefore three depth motion maps are generated for each depth video sequence. HOG features are extracted from each DMM and concatenated together to form a feature vector for the action.

Compared to the above depth features and recognition systems, histogram of oriented 4D normals (HON4D) outperforms them on all relevant benchmarks [119]. In HON4D, the depth sequence is considered as a 4D space of time, depth and spatial coordinates, and surface normal orientations are calculated for the 4D space. The 4D space is then quantized by a 600-cell polychoron, optimized, and projected into 120 bins, forming a 120-dimensional HON4D descriptor. The video sequence can also be divided into multiple Spatiotemporal cells in width, height and number of frames, and the HON4D descriptor is extracted separately from each cell. The depth sequence can then be represented as a concatenation of the HON4D descriptors. Due to the discriminative power of HON4D, it is used in our multimodal action recognition framework. Instead of extracting HON4D from the whole depth sequence, we apply it only for the hands, and extract the HON4D descriptors around the hand joints of each depth frame.

3.2 Classification

As introduced above, various features can be extracted from certain interest points or from the whole gesture. Therefore, there are several ways to represent a gesture sequence, which has an effect on the selection of suitable classification methods. In this section, we will give a brief introduction of the classification methods commonly used in the recognition

systems, and continue to have a detailed look into several methods that are applied in this thesis.

3.2.1 Methods for time series feature vectors

When features are extracted from each frame and stacked as a matrix to represent a gesture, they can be considered as a sequence of feature vectors, that is, a multidimensional time series. In general, the gestures consist of a different number of frames, and thus the sequences are also of different length. For this kind of features, the widely used classifers in action recognition systems include hidden Markov model (HMM) [124, 164, 91], conditional random field (CRF) [25], action graph [152, 82] and various variations of dynamic time warping (DTW) [32, 129]. As preprocessing, the directly extracted features are often clustered or quantized into keywords. The commonly used methods for clustering include K-means [177, 112, 164] and the Gaussian mixture model [25]. The centers of the clusters are used as the keywords or codewords. Therefore, each gesture is represented as a sequence of keywords with a varying length. According to a performance evaluation of HMM and DTW for gesture recognition from Kinect skeleton data [18], the DTW has advantages in term of accuracy and trivial requirements for the number of training samples.

Dynamic time warping

Dynamic time warping (DTW) is a well-known and popular algorithm to measure the similarity of signal sequences [104]. It is widely used e.g. in speech recognition and gesture recognition with skeletal data [127, 32, 1]. Let us define two sequences, $P = p_1, p_2, ..., p_N$, and $Q = q_1, q_2, ..., q_M$, where p_n and q_m are the signals at the nth and mth time index, respectively. A distance function is used to calculate the dissimilarity between the elements p_n and q_m from each sequence. The commonly used functions include the Euclidean and cosine distances. A cost matrix can then be constructed by calculating the dissimilarity distances between each pair of elements from the sequences. DTW finds the alignment between P and Q so that the sum of all distances between the aligned element pairs is minimized, which is also referred to as the minimum overall cost. The alignment with the minimum cost is denoted as the warping path, as it indicates the indices of the matched elements in the sequences. For basic DTW, the warping path has to fulfill two conditions: (1) the first and last elements of P and Q have to align to each other, that is $p_1 \leftrightarrow q_1$ and

$p_N \leftrightarrow q_M$; and (2) if $p_n \leftrightarrow q_m$, then $p_n \leftrightarrow q_{m+1}$ or $p_{n+1} \leftrightarrow q_m$ or $p_{n+1} \leftrightarrow q_{m+1}$, where the symbol $\leftrightarrow$ means that the left signal element is aligned or matched with the right signal element. To find the warping path under these constraints with computational efficiency, dynamic programming is often used. An example can be seen in Figure 3.6.

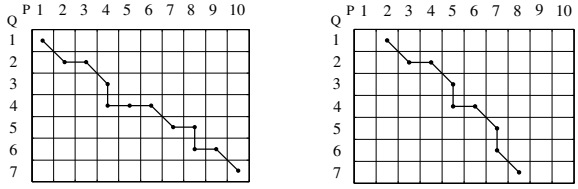


Figure 3.6. Example warping paths of DTW (left) and subsequence DTW (right). From Publication V.

There are many variants of DTW. Subsequence DTW (SS-DTW) is a variant that does not fulfill the first condition of basic DTW, that is, the beginning and end of the two sequences are not required to match each other. Instead, the longer sequence is assumed to contain a matching subsequence to the shorter sequence. Detailed information of DTW and SS-DTW can be found in Publication V.

3.2.2 Methods for features with a fixed dimensionality

Unlike gestures represented as time series signals, in some systems, each gesture is represented as a whole by a single feature vector with a fixed dimensionality. One strategy to obtain these fixed-dimensional features is to sample all gestures into a fixed number of frames. The features extracted from these frames can then be concatenated to form one feature vector, which usually becomes high-dimensional. Another strategy is to represent the gesture as a histogram. Features are extracted among the whole gesture and clustered into codewords. The gesture is then represented as a histogram of codewords.

When the gestures are represented by features with a fixed dimensionality, many pattern recognition methods, such as, K-nearest neighbors (KNN) [76], naive Bayes [168], and SVM [157], can be applied. KNN is one of the most straightforward methods for this task [41]. The distances between the feature of the testing gesture and the ones of the training gestures are calculated, and the gesture is classified to the class which appears most frequently among the K training features with the shortest distances. However, the standard brute-force searching method for KNN is not a good

solution especially for high-dimensional datasets. In order to improve the speed or memory requirement, approximate nearest neighbor (ANN) [60] algorithms provide a suitable replacement for KNN in some applications. As the name implies, ANN algorithms do not guarantee to return the actual nearest neighbor for each query but an approximate one [86]. The Fast Library for Approximate Nearest Neighbors (FLANN) [102, 103] is a very effective library for fast ANN searches in high-dimensional spaces. It contains two searching algorithms: hierarchical k-means trees with a priority search order and multiple randomized k-d trees. In Publications I and II, we have used FLANN to build multiple randomized k-d trees for the image descriptors in the server database.

Another method based on multiple trees, although serving a different functionality, is random forest [11, 55], which has had huge success in human pose recognition from depth data in real-time [135]. Since then it has also been applied in a gesture recognition system [177]. A random forest is an ensemble of decision trees. Each decision tree is constructed separately with a random selected subset of the training data, and contains internal nodes and leaf nodes. At each internal node, the data is split based on the output of a split function for a certain attribute value and a threhold. Each leaf node is labeled with a class or a probability distribution over the classes. During testing, each test sample proceeds down to one leaf node in each tree. The probability distributions associated with these leaf nodes are averaged over all trees. The class with the highest probability is assigned to the test sample. If probability distributions are not available but only a class label exists at each leaf node, then the class label with the maximum number of votes is assigned to the test data.

In our recognition system, we use extreme learning machine (ELM) [59] as our classifier. ELM has been succesfully used in many applications, including recognizing human activities from video data [100]. We compare ELM with three popular classifiers: *logistic regression* [9], *linear SVM with an approximate feature map* [151, 137] and *RBF-kernel SVM* [27, 145]. These three methods have their own advantages, as shown in the comparison. Logistic regression has a low computational complexity; RBF-kernel SVM is widely applied due to its powerfulness in classification, nevertheless with high computational costs; linear SVM with an approximate feature map performs in between the above two both in accuracy and computational cost. It simplifies the calculation of additive non-linear kernels (such as the χ^2 kernel) by approximation and uses a linear SVM as

the classifier, which results in much faster training and testing than with the original non-linear SVM. In the comparison, ELM performs favorably in both aspects of accuracy and computational cost. The comparison results can be found in Section 4.4.1 and Publication IV.

Extreme learning machine

The extreme learning machine (ELM) as an emerging algorithm can be applied for classification and regression [59, 57]. It is a single-hidden layer feedforward neural network (SLFN), but its core property is that the hidden layer in ELM does not need to be tuned. Figure 3.7 shows the structure of a single-hidden layer feedforward network with L hidden neurons.

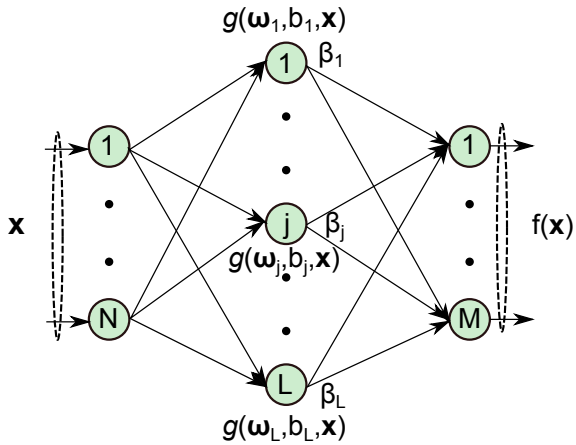


Figure 3.7. A single-hidden layer feedforward network.

Given P samples $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{P}$, where $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{iN}]^T \in \mathbb{R}^N$ and $\mathbf{y}_i = [y_{i1}, y_{i2}, \ldots, y_{iM}]^T \in \mathbb{R}^M$, the output function of standard SLFNs with L hidden neurons can be represented as

$$\mathbf{y}_i = f(\mathbf{x}_i) = \sum_{j=1}^{L} \boldsymbol{\beta}_j g(\boldsymbol{\omega}_j \cdot \mathbf{x}_i + b_j) \,, \tag{3.2}$$

where $\boldsymbol{\omega}_j = [\omega_{j1}, \omega_{j2}, \ldots, \omega_{jN}] \in \mathbb{R}^N$ is the input weight vector connecting the input layer to the jth hidden neuron, b_j is the bias of the jth hidden neuron, $g(\cdot)$ is a nonlinear piecewise continuous function, for example, sigmoid function and Gaussian function, and $\boldsymbol{\beta}_j = [\beta_{j1}, \beta_{j2}, \ldots \beta_{jM}]^T \in \mathbb{R}^M$ is the output weight vector connecting the jth hidden neuron and the output neurons. Equation 3.2 can be written compactly as

$$\mathbf{Y} = \mathbf{H}\boldsymbol{\beta} \,, \tag{3.3}$$

where the hidden layer output matrix $\mathbf{H}$ is

$$\mathbf{H} = \begin{bmatrix} g(\boldsymbol{\omega}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g(\boldsymbol{\omega}_L \cdot \mathbf{x}_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(\boldsymbol{\omega}_1 \cdot \mathbf{x}_P + b_1) & \cdots & g(\boldsymbol{\omega}_L \cdot \mathbf{x}_P + b_L) \end{bmatrix}_{P \times L}, \tag{3.4}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \boldsymbol{\beta}_2^T \\ \cdots \\ \boldsymbol{\beta}_L^T \end{bmatrix}_{L \times M} \quad \text{and} \quad \mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^T \\ \mathbf{y}_2^T \\ \cdots \\ \mathbf{y}_P^T \end{bmatrix}_{P \times M} \tag{3.5}$$

Traditionally, the values of the input weight vector $\boldsymbol{\omega}_j$, bias b_j and the output weight vector $\boldsymbol{\beta}_j$ for each hidden node are learned through iterative processing, with backpropagation [15] being the most popular learning algorithm used in feedforward neural networks. Unlike the traditional neural networks, in the extreme learning machine the values for $\boldsymbol{\omega}_j$ and b_j are not learnt, but instead are assigned random values that remain fixed. Training an ELM simply equals to finding a least-squares solution $\hat{\boldsymbol{\beta}}$ to Equation 3.3. If the number of hidden neurons L is equal to the number of training samples P, matrix $\mathbf{H}$ is square and invertible. However, in most cases $L \ll P$, and the smallest norm least-squares solution of the linear system $\mathbf{H}\boldsymbol{\beta} = \mathbf{Y}$ can be obtained by

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{Y}\,, \tag{3.6}$$

where $\mathbf{H}^{\dagger}$ is the Moore-Penrose pseudoinverse of matrix $\mathbf{H}$. Several methods exist to calculate $\mathbf{H}^{\dagger}$, including the orthogonal projection method and singular value decomposition (SVD) [126]. The solution $\hat{\boldsymbol{\beta}}$ gives the minimum training error for the linear system, provides the smallest norm of weights, and is unique.

In summary, for ELM, the input weights and the hidden layer biases do not need to be learned, resulting in the learning of ELM being extremely fast. A comprehensive study and comparisons between ELM and SVM show that ELM can achieve a better generalization performance for multiclass classification together with much faster learning than traditional SVM [57]. In addition to the above introduced basic ELM, there are also many ELM variants, such as kernel-based ELM, incremental ELM, and so on [58]. Parallelized ELM with GPUs [150] can make ELM even more feasible for large scale datasets. In our gesture recognition system, we adopt the basic ELM, which facilitates the recognition to be done in real-time.

4. Skeleton-based action recognition

The previous chapters provide an overview of the current existing gesture recognition systems and some methodologies used in our work. In this chapter, we will present our action recognition system for skeleton data as a whole. In the next chapter, we will demonstrate our system on experiments on multimodal data.

4.1 Overview of the recognition system

Figure 4.1 gives a graphical overview of the recognition system developed in this thesis. The input for the system is motion sequences, generated either by a motion capture system or an RGB-D sensor. The system consists of two parts: feature extraction and action classification. The former extracts different features for each frame of the sequence, and the latter classifies the features by extreme learning machine, and models the overall outputs of ELM on the sequence level. As a result, the final classification result for the action is obtained.

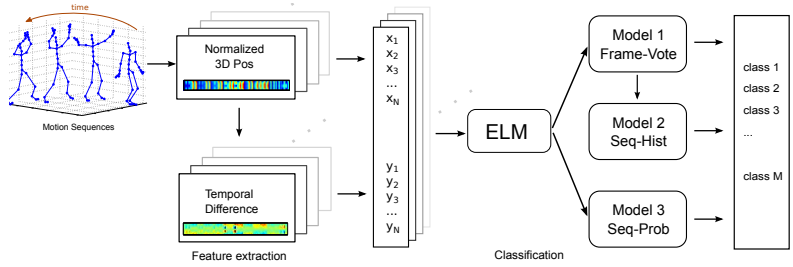


Figure 4.1. An overview of the action recognition system. From Publication IV.

4.2 Skeletal features

Simple and effective features are essential for an accurate recognition system. In the feature extraction part, we propose skeletal features which capture the spatial relational information between the joints and the temporal relational information within the gesture. Furthermore we visualize our features on a two-dimensional space using a deep auto-encoder for qualitative analysis.

4.2.1 Feature extraction

Normalized 3D joint positions (NP)

The skeleton model is a constitution of joints represented by 3D coordinates, which contains very rich raw information about the posture. However, the joint coordinates are closely related to the circumstances in which the skeleton model is generated. The coordinate system varies in uncontrolled recording environments, which directly influences the joint coordinate values. Next, even in the same coordinate system, multiple instances of the same gesture performed by the same actor are likely to have different coordinate values due to translation and rotation. Moreover, attributable to the different body sizes of the performers, the same gesture by different performers will have different coordinate representations. Therefore in order to directly use the 3D coordinates, it is essential to register the joint coordinates into a common coordinate system.

The sources of the skeleton model are mainly the data from a motion capture system and RGB-D sensors. In addition to providing the 3D coordinates of the joints, the former also provides rotational and translational information of the joints in relation to the other joints, which is not often available for the latter. In order to make these skeleton coordinates comparable, all skeletons are rotated into the same orientation and the root joint of the skeleton is translated to the origin, which causes the coordinates of the hip joints of the same actor to overlap regardless of the posture (the root joint can be seen in Figure 4.8). For example, Figure 4.2 shows sampled frames from a “cartwheel” action from the HDM05 database in the original coordinates and registered coordinates after the transformation. For mocap data, a straightforward way exists to transform the coordinates by setting the rotation matrix and the translation vector of the root joint

to identity and zero. This method is also used in [73] for motion capture data retrieval.

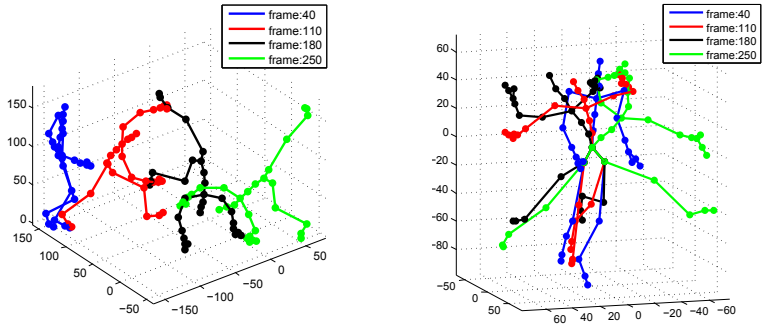


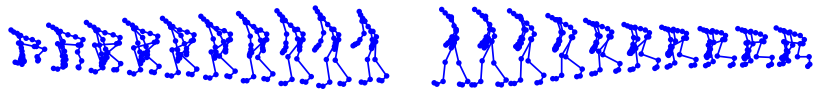
Figure 4.2. A "cartwheel" action drawn in original (left) and transformed (right) coordinates. Adapted from Publication III.

For skeletons extracted with a RGB-D sensor, usually only the 3D joint coordinates are available without the rotation and translation information. To transform all skeletons into the same orientation and to translate the root joints to the origin, one skeleton can be selected as a common basis, with its orientation considered as a reference for the other skeletons to be transformed into. In our method, this reference skeleton is selected randomly. As seen in Figure 4.2, after the transformation, the hips and the root joint of each skeleton overlap correspondingly and the planes formed by them also overlap with each other. This inspires the solution of the transformation of the RGB-D skeletons. We translate the roots of all skeletons to origin, then rotate the skeletons so that the planes formed by hips and root overlap with the one of the reference skeleton, with the condition that the sum of the distances between the transformed hip joints to the corresponding hip joints in the reference skeleton is minimized. A more detailed description can be found in Publication III. The coordinates are normalized to be invariant to the size of the performer by normalizing the sum of the distances of the connected joints to one. Finally, the feature vector is represented as the concatenation of the joint coordinates.

Temporal difference of feature vectors (TD)

The above feature captures the characteristics of a single posture. However, in general, a gesture is composed of multiple postures or frames, and the relations between the postures carries vital distinctive information of the gesture. One extreme case is when two actions are kinematically inverse to each other. Figure 4.3 shows two such actions: *StandUpKnee* and *SitDownKnee*. Here, we can see that for each posture in one gesture there

is an almost identical counterpart in the other one. A classifier can easily misclassify the features extracted only from each posture. Therefore, it is necessary to take the temporal order of the postures into account to compensate for the weakness of the features extracted on frame level in this aspect.



(a) **(b)**

Figure 4.3. One pair of inverse actions. (a) StandUpKnee; (b) SitDownKnee. From Publication III.

To capture the temporal relationship among postures within a gesture, one method is to subtract the features of one frame from another to represent the differences related with the chronological order. Assuming the skeletal feature vector of the kth frame in a gesture sequence with K frames is $\mathbf{x}_k^d$, the temporal difference of feature vectors $\mathbf{x}_k^{td}$ can be calculated as

$$\mathbf{x}_k^{td} = \begin{cases} \mathbf{x}_k^d & 1 \leq k < k' \\ \dfrac{\mathbf{x}_k^d - \mathbf{x}_{k-k'+1}^d}{||\mathbf{x}_k^d - \mathbf{x}_{k-k'+1}^d||} & k' \leq k \leq K \end{cases} \tag{4.1}$$

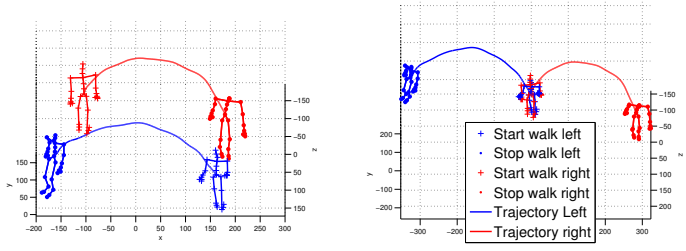
where k' is the temporal offset parameter, $1 < k' < K$.

After obtaining $\mathbf{x}_k^{td}$, it can be concatenated with $\mathbf{x}_k^d$ to form a new feature which combines the distinctive information of the skeleton for a single frame and the relation between that frame with an earlier frame in the gesture. The final feature vector can be written as $\mathbf{x} = [\,(\mathbf{x}^d)^T\ (\mathbf{x}^{td})^T\,]^T$. The dimensionality of the feature depends on the number of joints used in the original skeletal feature. Based on the characteristics of gestures, not all joints may be necessary to be used in skeletal feature. For example, for sign language gestures, the lower body of the actor often remains still or even invisible and mostly only the arms and hands move. In this case, the joints of the lower body provide the same information for all gestures and can therefore be excluded from the skeletal feature. For example with the NP feature and assuming n_j joints are selected, the dimensionality of $\mathbf{x}$ is $n = 2 \cdot 3 \cdot n_j$. In the rest of the thesis, when we extract frame-level skeleton features, we always concatenate the original feature and the temporal difference vector to form the final skeletal feature to be classified in the system, without always explicitly stating this.

Normalized trajectory (NT)

As stated in the previous section, the extraction of the NP feature requires translation of the roots of the skeletons to the same position. This procedure eliminates the absolute movement of the gesture, as Figure 4.2 illustrates. However, the absolute movement possibly carries information related to the differentiation between the gestures, and therefore the gesture trajectory feature is calculated and named as normalized trajectory. It is generated by translating the coordinates of the root joint of the first frame to the origin of the coordinates. Correspondingly, all the coordinates of the other joints are translated to maintain the original geometric relation. After the translation, new coordinates are assigned to all the joints. Then these coordinates are normalized into $[-1,\ 1]$. The NT feature corresponds to the coordinates of the root joint after translation and normalization for each frame. The NT feature can then be concatenated with the NP+TD feature to form a new feature vector. We refer explicitly to this feature as NP+TD+NT in the rest of the thesis. The detailed calculation can be found in Publication VI.

Figure 4.4 shows an intuitive example of the effect of the NT feature. Figure 4.4(a) shows two motions: *walk in a left circle* and *walk in a right circle*. In this figure, the trajectories of the left and right circle are not distinguishable. Figure 4.4(b) shows the trajectories after the transformation.



(a) (b)

Figure 4.4. Trajectories of two different walks in (a) original and (b) transformed coordinates. From Publication VI.

There are multiple ways to extract features from a skeleton model. In addition to the NP feature, we also experiment with several other features, combined with the concept of temporal differencing.

Pairwise distance vector (PW)

The pairwise distance vector calculates all distances between the selected joints, normalized so that the sum of all elements equals to one. It is a

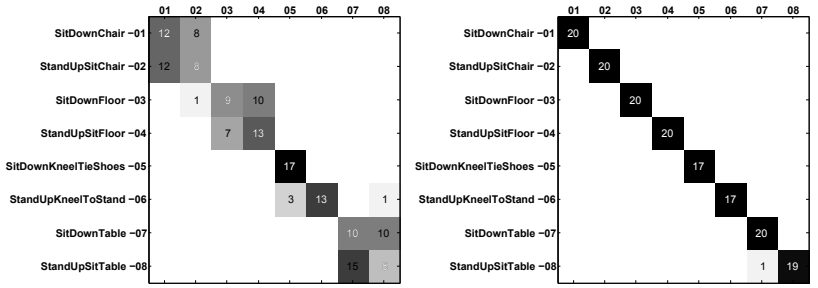
commonly used feature [168, 177]. It is similar to the NP feature in the sense that both of them try to obtain the relations between joints. The NP feature uses the transformed joint coordinates whereas the PW feature directly utilizes the distances between the joints. In general, the former carries more information but the latter is easier to calculate.

Centroid distance vector (CEN)
Another way to extract skeletal features is to use the centroid of the skeleton. Here we consider the centroid of the triangle formed by the neck and hips as the centroid of the body. The elements of the CEN feature vector consist of the distances between the joints and the centroid, normalized by the sum of the distances.

Key joints distance vector (KEY)
Similar to the centroid distance vector, the key joints distance vector is calculated as the distances between a set of key joints and the other joints. In a later experiment the following three key joints are used: *head*, *root* and *left knee*.

4.2.2 Effects and parameters of TD feature

To observe the effect of the temporal difference of feature vectors, we record recognition accuracies with and without the TD feature on a selection of gestures from the HDM05 database. In total, eight gestures with 154 instances are selected. The gestures form four pairs with the actions in reverse chronological order, as for example, in *SitDownChair* and *StandUpChair*. The gestures are listed in Figure 4.5.



(a) **(b)**

Figure 4.5. Confusion matrix for four pairs of inverse actions. (a) Using only NP features; (b) using concatenated features NP+TD. Adapted from Publication III.

In the experiment, the classifier and modeling methods are the same in both cases except for the used feature which is with and without the TD components. The confusion matrices for the experiment are shown in Figure 4.5. We can see that with only the NP feature, the gestures in each pair are often misclassified to each other, whereas after concatenating with the temporal difference feature vector, the gestures are clearly distinguished from each other. The average accuracy is increased from 56.49% to 99.35%.

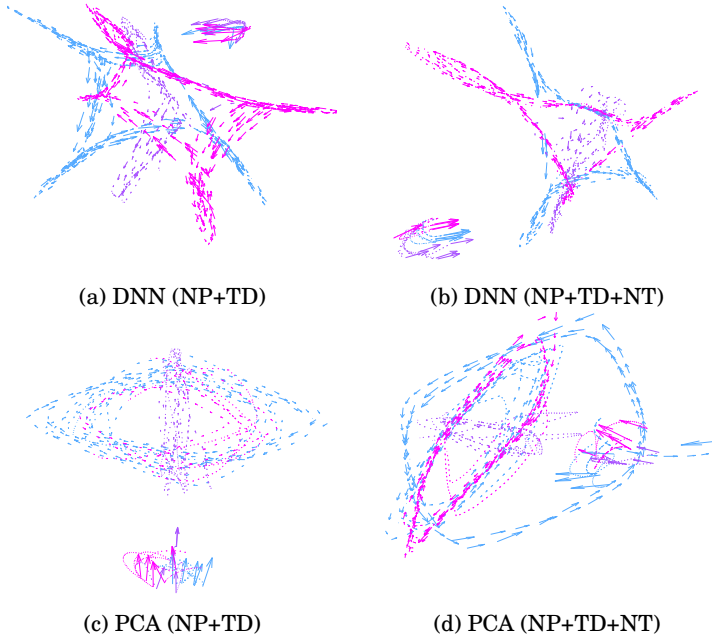
One parameter required in the TD feature is the temporal offset. Through experiments, a difference of 0.3 seconds between the frames has been observed to be a robust value. Therefore we use this value for all the remaining experiments. Detailed information about the parameter selection can be found in Publication III.

4.2.3 Visualization of the features

In order to visualize the features, the dimensionality also needs to be reduced to two or three dimensions. PCA is often applied in many applications [152, 168] for this purpose. The dimensionality of the NP+TD features is six times the number of joints used in the skeleton, which is often no more than 60 dimensions. This is not a burden for our classifier, but in order to have an intuitive understanding and observation of the effectiveness of our features, we reduce the dimensionality to two to observe them in a 2-D image.

We apply both the commonly used PCA and a deep autoencoder [54] to perform the dimensionality reduction. For PCA, the two largest principal components are used to visualize the feature. The deep autoencoder has two linear neurons in the middle layer and three hidden layers of size 1000, 500 and 100 between the input and the middle layers. No label information is used to train the deep autoencoders. We visualize the features with and without the normalized trajectories (NT) to see what the relative feature (NP+TD) provides to the system and the impact of the absolute feature by both PCA and the autoencoder.

We first visualize three distinct but very similar actions: *rotateArmsRBackward*, *rotateArmsBothBackward* and *rotateArmsLBackward*. As Figure 4.6 shows, these gestures are clearly distinguishable in the two dimensional space when the deep autoencoder is applied. However, *rotateArmsRBackward* and *rotateArmsLBackward* are visually not distinguishable at all by PCA when only the NP and TD features are used (see



(a) DNN (NP+TD) (b) DNN (NP+TD+NT)

(c) PCA (NP+TD) (d) PCA (NP+TD+NT)

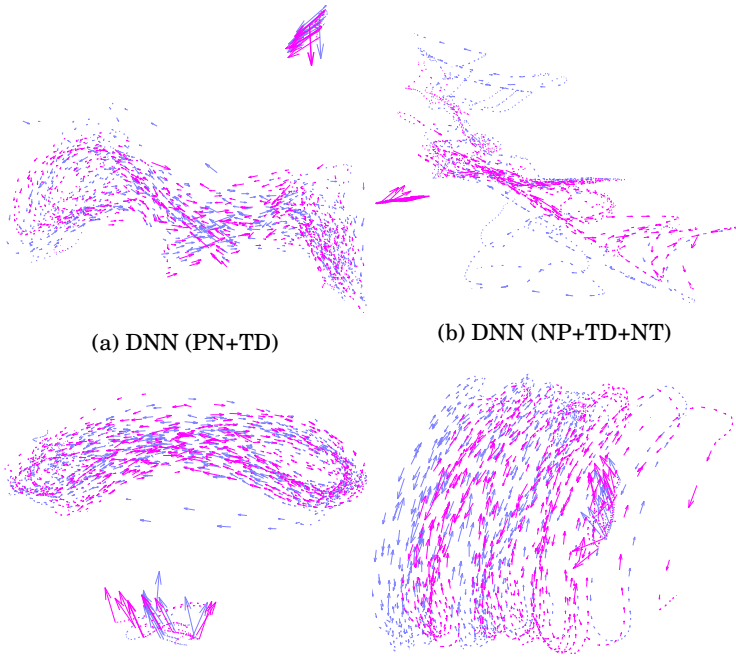
Figure 4.6. Visualization of actions *rotateArmsRBackward* (blue), *rotateArmsBothBackward* (purple) and *rotateArmsLBackward* (red). Each arrow denotes the direction and magnitude of change in the latent space. Five randomly selected sequences per gesture are shown. From Publication VI.

Figure 4.6 (c)). Even with all three features (NP+TD+NT), using PCA does not clearly distinguish the features.

Figure 4.7 shows the visualization of two gestures, *jogLeftCircle* and *jogRightCircle*. When only NP and TD features are used, neither the deep autoencoder nor PCA is able to capture the differences between the gestures. However, the deep autoencoder can distinguish these gestures clearly when all three features are used (see Figure 4.7 (b)), whereas PCA does not capture the difference. The comparison shows that a deep neural network with multiple nonlinear hidden layers can learn a more discriminative structure of the data. More detailed information about the influence of the NT feature on the recognition can be found in Publication VI.

4.3 Mocap vs Kinect

The skeletons from different sources are often in different formats, as illustrated in Figure 4.8. The leftmost skeleton is of the format used in the CMU Graphics Lab Motion Capture Database [26] and the Motion



(a) DNN (PN+TD) (b) DNN (NP+TD+NT)

(c) PCA (NP+TD) (d) PCA (NP+TD+NT)

Figure 4.7. Visualization of actions *jogLeftCircle* (blue) and *jogRightCircle* (red). Each arrow denotes the direction and magnitude of change in the latent space. Ten randomly chosen sequences per action were visualized. From Publication VI.

Capture Database HDM05 [107] with 31 joints. The other two skeletons are extracted from Kinect data but by different software. The middle skeleton is from the Microsoft Kinect SDK [97] and has 20 joints. The rightmost skeleton is from the PrimeSense Natural Interaction Middleware (NiTE) [118] with 15 joints. Even though the number of joints differ, the essential joints are available in all formats. These include the hands, feet, elbows, knees and root.

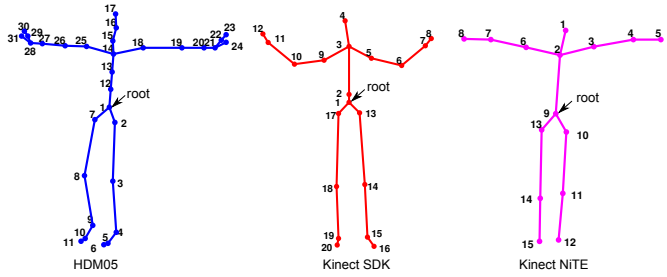


Figure 4.8. Skeleton models from different sources. From Publication IV.

4.3.1 Recognition performance

Due to the mobility and low cost of the RGB-D sensor, it can replace the motion capture system for certain functionalities. Therefore it is useful to analyze the performance of the proposed gesture recognition method with the skeletons generated both from an RGB-D sensor and from mocap, which leads to a reasonable selection of systems to generate the desired skeleton data.

To compare the performance of the different skeleton sources, we use the same features and settings in the recognition system. Furthermore, in order to compare our system's performance with the results published in [152], we use the same mocap data as in [152], which consists of 10 classes of gestures and a total of 156 instances from the HDM05 database. We also record the same gestures on our own with Kinect using the NiTE middleware. In this Kinect dataset, the number of actors and the number of instances performed by each actor is exactly the same as in the mocap dataset. Figure 4.9 shows as an example the RGB-D images and the corresponding skeleton.

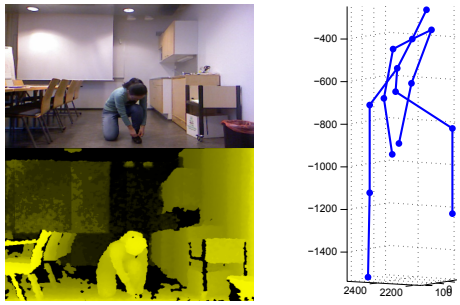


Figure 4.9. RGB and depth images (left), and the corresponding skeleton (right). From Publication III.

In this experiment we select two sets of joints: all 15 joints in the NiTE skeleton format and a reduced set of five joints (head, hands and feet). By using the two sets of joints, we can see the influences of the number of joints for the recognition performance. The four kinds of features described in Section 4.2.1 are extracted for both sets of joints. The average accuracies of the 10 classes are shown in Table 4.1, with the dimensionality of each feature is shown in parentheses. Detailed recognition accuracy information for each class can be found in Publication III.

From Table 4.1 we can see that for the mocap data our system provides higher classification accuracies with all features than the result provided

Table 4.1. Classification accuracy of 10 gesture classes from both mocap and Kinect data. Adapted from Publication III.

Accuracy	15 joints				5 joints				
	np (90)	pw (210)	cen (30)	key (78)	np (30)	pw (20)	cen (10)	key (28)	[152] (%)
Mocap(%)	100	100	98.8	99.3	99.4	99.5	99.5	100	90.9
Kinect(%)	96.1	95.0	83.1	92.5	94.7	92.0	83.3	90.7	

in [152], which uses pairwise distance matrices and an action graph for the classification. We can also observe that the accuracies from mocap are all higher than the corresponding Kinect ones. This was to be expected and can be explained by the unstability and noisiness of the Kinect skeletons. For example, Figure 4.9 shows the right knee of the skeleton not bent as in the corresponding RGB-D image. However, the accuracy with Kinect is only about 5% lower than mocap for the NP feature in this dataset.

The dimensionality of the features varies based on the number of joints used in the feature. With the reduced set of joints, the accuracy remains high but the dimensionality has reduced significantly. Therefore the used subset of joints, that is, head, hands and feet, is a good selection of primary joints in this setting. The performance of the four features is more or less the same for mocap data; for Kinect data, the NP feature is the best for both sets of joints. NP performs a little better than the pairwise distance, and both provide much better accuracies than the other two features. In fact, the NP and PW features are commonly adopted into our recognition system.

4.3.2 Gesture alignment

In the above subsection, we have compared the performance of skeletons from an RGB-D sensor and a motion capture system within the same gesture recognition system. The experiments indirectly show the noisiness and unstability of the RGB-D skeleton compared to mocap skeletons. However, if a quantitative evaluation of the RGD-D skeleton is to be conducted, the groundtruth of the skeletons are required for the corresponding skeletons. In [87], a wooden stick is used as groundtruth measurement for Kinect skeletons. As the mocap skeleton is the most precisely generated one, it is undoubtly the best option for the groundtruth skeleton. However, the RGB-D sensor and motion capture are two independent systems, and

therefore the hardware synchronization of the systems is not trivial to accomplish. The time stamps of the skeletons are not available in almost all of the datasets. The skeletons of a gesture can be considered as a time series signal, and therefore we use dynamic time warping to find the correspondences between the RGB-D and the mocap skeletons for the same gestures.

Data acquisition

To record the skeleton gestures with the two kinds of systems, an OptiTrack motion capture system and one Kinect device are used in the data collection. No calibration between these two systems is performed. When an actor performs a gesture, the two systems are started to record separately with the mocap always started earlier and ended later than the Kinect. A total of six gesture sequences are recorded and each one is named according to the main action in the video as *jump*, *sit*, *stand & walk*, *turn*, *walk* and *wave hand*. NiTE is used to generate the skeleton for Kinect with its 15-joints format, and the mocap skeleton has 20 joints.

Feature extraction

To convert the skeleton model into time series signals, we need to extract suitable features from the skeletons. The two kinds of skeletons have different numbers of joints, so we first need to convert them into the same structure, that is, to have the same set of joints. Since the mocap skeleton has more joints than the Kinect one, the extra joints are deleted from the mocap skeletons. The details of the skeleton simplification method can be found in Publication V.

We use the centroid distance vector (CEN) described in Section 4.2.1. It is a normalized 15-dimensional feature vector due to the simplified skeleton having 15 joints. It should be noted that the CEN feature is not the only option for the alignment. As long as the feature can capture the skeletal information in a vector format, it can be used in our method. However, the dimensionality of the feature vector influences the computational complexity, which should be considered as one of the key factors for selecting the feature for the DTW method.

Alignment of the skeleton sequences

During the recording, the frame rate of mocap is accurately 100 fps and of Kinect theoretically 30 fps, but in practice some frames are missing, which makes the frame rate of Kinect not accurate. We also manually

control the recording so that the mocap recording starts earlier and ends later than Kinect, which makes it possible to use SS-DTW to align the skeletons. Considering both the difference of the frame rates and the missing frames in the Kinect recordings, we need to modify the step size of SS-DTW (Section 3.2.1) to fulfill our requirements.

Figure 4.10 illustrates the required step modification for the alignment. Let us assume the Kinect frame f_i^k matches with the mocap frame f_j^m. Theorectically, the next Kinect frame will match the mocap frames between the 3rd and 4th frame. However, this assumption can rarely be true, because of the uncertainty of the synchronization and the frame rate ratio of 100 to 30, therefore there is inaccuracy of this assumed alignment. Taking the inaccuracy of the assumed alignment, the matched mocap frames lie from f_{j+3}^m to f_{j+5}^m corresponding to the Kinect frame f_{i+1}^k. Considering the possibility of one or two frames missing in the Kinect recording, the possible matching frames can be extended to f_{j+11}^m. Overall, the next possible matching frame ranges from the 3rd to the 11th frame in the mocap sequence. By replacing the step condition in the accumulated cost matrix of SS-DTW with the varied step condition, a best warping path between Kinect and mocap skeleton can be obtained.

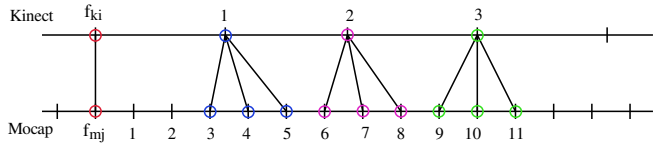


Figure 4.10. The matching pattern between Kinect and motion capture skeleton. From Publication V.

Evaluation of the alignment

Figure 4.11 shows one example of aligned skeleton sequences by SS-DTW and three other methods. The ill-matched skeletons are marked by ellipses or by a rectangle. We can see that the alignment by SS-DTW is visually very similar to the Kinect sequence. Detailed information of the other methods can be found in Publication V.

The aligned sequences are also evaluated quantitatively by measuring the minimum overall distance either between the feature vectors or between the transformed skeletons. The former calculates the Euclidean distance of the extracted features from the matched skeletons and sums all distances over the whole aligned sequences. The latter tries to overlap

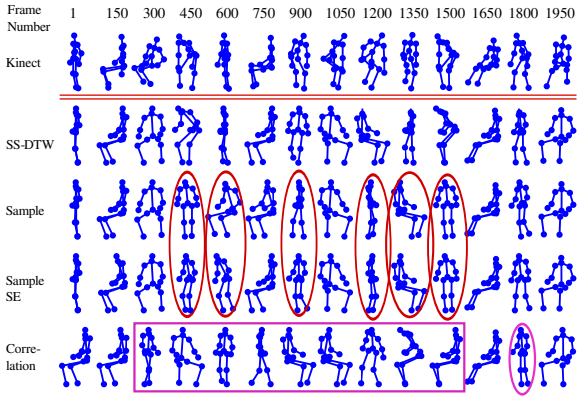


Figure 4.11. Skeletons aligned with different methods. From Publication V.

the two matched skeletons to calculate the distances between the corresponding joints. The overlapping of the skeletons from the two different coordinate systems can be achieved similarly as in the calculation of the NP feature. All Kinect skeletons are transformed to overlap the matched mocap skeletons. The distances between the corresponding joints are summed together over the whole aligned sequences. The evaluation measure is the sum of all Euclidean distances between the corresponding joints for each frame in the aligned sequence.

The results for all sequences with different alignment methods by these two evaluation methods are shown in Figure 4.12. In all cases, the distances by the SS-DTW alignment are smaller than with the other methods. The distance measuring methods can also be used to evaluate RGB-D skeletons generated by different algorithms with a groundtruth skeleton.

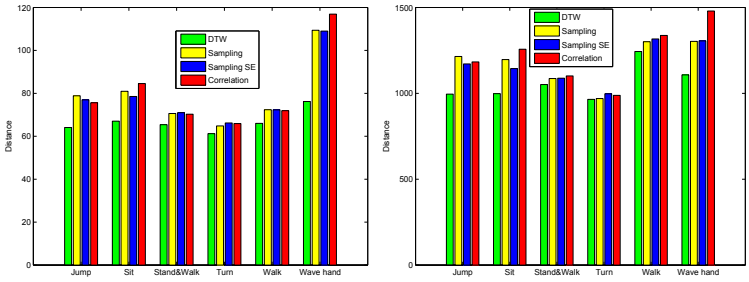


Figure 4.12. The distances of feature vector (left) and of skeleton coordinates (right) for all recordings. Adapted from Publication V.

4.4 Classification and modeling

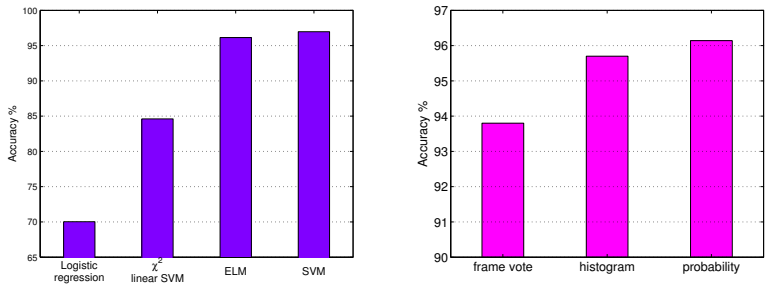
After the features are extracted from the skeleton data, the next step is to classify the features to recognize the gestures. As Figure 4.1 illustrates, our classification method can be divided into two phases: frame-level classification and sequence-level modeling. The former is to classify the features frame by frame; the latter is to use all the outputs from the frame-level classifiers to evaluate the final classification result for the gesture. In this modular design, the length of the gesture is not a limiting factor for feature extraction. When the features are extracted based on each frame, more detailed information can be preserved in the features. Also, the classification can be performed immediately as each frame arrives, which leaves only the light computation of the sequence level model to the end of the gesture. This design greatly facilitates real-time recognition of the gestures.

4.4.1 Frame-level classification

In addition to the accuracy of the classifier, the computational complexity is a critical factor for the feasibility of our recognition system for real-time applications. In our system we propose to use extreme learning machines to classify the features. We have compared the ELM with several other popular classifiers. The three classifiers used in the comparison were *logistic regression*, a *linear SVM with an approximate feature map* [151, 137] and *RBF-kernel SVM*. For ELM, we use 750 hidden neurons. Detailed information of the selection of the number of neurons can be found in Publication IV.

In the experiment, we use 40 classes of gestures with 790 instances from the HDM05 dataset. The list of gestures can be found in the Appendix A of Publication IV. Except for the classifiers, all other settings in the recognition system remain unchanged. The average classification accuracies over the 40 classes are shown in Figure 4.13(a). We can see that both the ELM and RBF-kernel SVM reach an accuracy of 96%. The other two classifiers have much lower accuracies in this experiment.

On the other hand, the training and testing times for all classifiers are shown in Table 4.2. The testing time is the average value for a gesture instance which includes both phases of frame-level classification and sequence-level modeling (Section 4.4.2). The training dataset contains about 200 000 features. All experiments are conducted on a Intel(R)



(a) **(b)**

Figure 4.13. Comparisons of classifiers and post-learning modeling methods. (a) Accuracies of different classifiers; (b) accuracies of different sequence modeling methods. Adapted from Publication IV.

Xeon(R) CPU at 3.3 GHz and 16 GB of memory. Table 4.2 shows that it takes 4.7 milliseconds on average for the ELM to classify one gesture whereas the RBF-kernel SVM requires over 500 times more time. The training takes 30 seconds and 21 minutes for ELM and SVM, respectively. Therefore, ELM can be considered as a good tradeoff between accuracy and computational complexity in frame-level classification.

Table 4.2. Training and testing times for different classifiers. From Publication IV.

Classifier	Logistic regression	χ^2 linear SVM	ELM	SVM
Testing Time (ms)	0.88	2.7	4.7	2500
Training Time (s)	43	110	31	1300

4.4.2 Sequence level modeling

The output of ELM as in Equation 3.2 is a vector in which each element reflects the similarity to the corresponding class. In our recognition system, all frames of the gesture are classified first, and then the frame-level outputs are utilized to obtain the final classification result for the gesture.

Assume t is a test gesture consisting of the features for each frame in the gesture as $t = \{\mathbf{x}_1, \ldots, \mathbf{x}_q, \ldots, \mathbf{x}_Q\}$, where $\mathbf{x}_q$ is a feature vector for frame q and Q is the number of frames in the gesture. The whole output of frame-level classification for the gesture t can be written in a matrix form

as

$$\mathbf{\Omega} = \begin{bmatrix} \hat{\mathbf{c}}_1 \\ \hat{\mathbf{c}}_2 \\ \vdots \\ \hat{\mathbf{c}}_q \\ \vdots \\ \hat{\mathbf{c}}_Q \end{bmatrix} = \begin{bmatrix} \hat{c}_{1,1} & \hat{c}_{1,2} & \dots & \hat{c}_{1,m} & \dots & \hat{c}_{1,M} \\ \hat{c}_{2,1} & \hat{c}_{2,2} & \dots & \hat{c}_{2,m} & \dots & \hat{c}_{2,M} \\ \vdots & & & \vdots & & \vdots \\ \hat{c}_{q,1} & \hat{c}_{q,2} & \dots & \hat{c}_{q,m} & \dots & \hat{c}_{q,M} \\ \vdots & & & \vdots & & \vdots \\ \hat{c}_{Q,1} & \hat{c}_{Q,2} & \dots & \hat{c}_{Q,m} & \dots & \hat{c}_{Q,M} \end{bmatrix}, \tag{4.2}$$

where $\hat{c}_{q,m}$ is the output of the classifier for class m from the qth frame feature. We continue to process the output matrix $\mathbf{\Omega}$ to obtain the final classification vector $\hat{\mathbf{y}}$, where $\hat{\mathbf{y}} = [\hat{y}_1 \ \dots \ \hat{y}_m \ \dots \ \hat{y}_M]$, $\hat{y}_m \in \{0, 1\}$, $1 \leq m \leq M$ and $\sum_{i=1}^{M} \hat{y}_i = 1$. If $\hat{y}_m = 1$, t is classified to the class A_m. We use the following three methods to obtain $\hat{\mathbf{y}}$ from $\mathbf{\Omega}$.

Frame-wise voting model

In this model, each frame q is first classified to the class which has the largest output value in $\hat{\mathbf{c}}_q$. The number of frames is then counted for each class, and the final class of gesture t is obtained using majority voting. This method does not require the output of the classifier to be a vector with numerical values. Rather, as long as the class of each frame is available, majority voting can be used to calculate the final class of the gesture.

Sequence histogram model

In this model, each gesture is expressed as a normalized histogram. We first build histogram models for each class from the training data, which leads to M-bin histograms. The L1 norm distance is then calculated between the test gesture histogram and the histogram of each class. The test gesture is classified to the class whose histogram has the minimum distance to the histogram of test gesture.

In order to obtain the histograms, we first count the number of frames classified into each class in a gesture as in the frame-wise voting model. Any sequence s can then be expressed by a normalized histogram as

$$\mathbf{h}_s = \frac{1}{Q} [u_1 \ \dots \ u_m \ \dots \ u_M], \tag{4.3}$$

where u_m is the number of frames classified as class m, and Q is the total number of frames in the gesture.

After training the ELM, we can classify the training gestures to obtain the histogram models as above. For the L_m training sequences $\{s_m^1, ..., s_m^j, ..., s_m^{L_m}\}$ belonging to the action A_m, the corresponding histograms are $H_m = \{\mathbf{h}_m^1, ..., \mathbf{h}_m^j, ..., \mathbf{h}_m^{L_m}\}$. By averaging the histograms

in H_m, the class A_m can be represented by the histogram

$$\mathbf{h}_m = \frac{\sum_{j=1}^{L_m} \mathbf{h}_m^j}{L_m} \ . \tag{4.4}$$

As a result, a total of M histogram models are built from the training data, each corresponding to one gesture class. For a test sequence t, the normalized histogram $\mathbf{h}_t$ is obtained as in Equation 4.3, and the distance between $\mathbf{h}_t$ and $\mathbf{h}_m$ can be calculated as

$$d_m = \|\mathbf{h}_t - \mathbf{h}_m\|_1 \ . \tag{4.5}$$

By calculating the distances between the test sequence histogram and all class histograms, the test sequence is classified to the class with the minimum distance.

Sequence probability model

We can also model the posterior distribution of classes given each frame. If the sequence t belongs to an action A_m, every frame in the sequence also belongs to A_m. Therefore we use the joint probability of all frames in a gesture to determine the class of the gesture. We convert the outputs $\hat{c}_{q,m}$ into probabilities with the logistic sigmoid function

$$p(\hat{y}_m = 1|\mathbf{x}_q) = \frac{1}{1 + \exp(-\gamma \hat{c}_{q,m})} \ , \tag{4.6}$$

where γ is the slope of the logistic sigmoid. Its value can be determined by using validation data.

The joint probability of the test gesture t is

$$p(\hat{y}_m = 1 \mid t) = p(\hat{y}_m = 1 \mid \mathbf{x}_1, \ldots, \mathbf{x}_q, \ldots, \mathbf{x}_Q) \ . \tag{4.7}$$

If we assume temporal independence among the frames in a sequence, which means that the class of each frame depends only on the features of that frame, Equation 4.7 can be simplified into

$$p(\hat{y}_m = 1 \mid t) = \prod_{q=1}^{Q} p(\hat{y}_m = 1 \mid \mathbf{x}_q). \tag{4.8}$$

The sequence t is then classified into the class with the largest joint probability. In reality, the frames are not independent. Through empirical verification we observed that the weighted arithmetic mean

$$d_m = \sum_{q=1}^{Q} w_q \, p(\hat{y}_m = 1 \mid \mathbf{x}_q), \tag{4.9}$$

often provides better accuracy. The weights w_q are obtained from a normalized Gaussian distribution, $w_q = \frac{1}{Z}\mathcal{N}(q; \frac{Q}{2}, \sigma^2)$, normalized so that

$\sum_{q=1}^{Q} w_q = 1$. The Gaussian function applied here is used to lessen the influence of the beginning and ending frames and to give more weight on the frames in the middle of the gesture. This is because the start and end of a gesture tend to be more similar among different gestures and the middle parts contain more distinctive information. Finally, we classify a test sequence to the class m with the maximum value for d_m.

Comparison of the three modeling methods

In order to study the performance of these three modeling methods, we performed experiments on the HDM5 dataset as in Section 4.4.1. After the frame-level classification, we used the same outputs from ELM as input for the three models. The classification accuracies of the different models can be seen in Figure 4.13(b). The sequence probability model obtained the highest accuracy, slightly outperforming the sequence histogram model. The frame-wise voting model was clearly inferior to the other two models. Therefore, we primarily use the sequence probability model in our experiments.

4.5 Other application examples

In addition to the HDM05 dataset, we also test our system on two other popular public datasets which both include skeleton data.

4.5.1 Microsoft Research Cambridge-12 Kinect gesture data set

The Microsoft Research Cambridge-12 (MSRC-12) Kinect gesture data set was collected for the evaluation of different ways of instructions given to performers for recording gestures [39]. In the dataset, five different kinds of instructions were given to the performers for conducting the same kind of actions. In total, 12 actions were collected for each instruction. Due to the five different instruction types, the whole dataset can be divided into five groups respectively. We use the same data group as in [25], in which the performers were given the instructions from videos played on a screen in front of them. The gestures were performed by a total of 30 people, and the same gesture was performed repeatly and recorded as a stream into one sequence. The data collector also specified the frame indices to mark the middle points of the gestures in the sequence. We cut the sequences containing multiple action instances into multiple files based on the middle

point indices, so that each file is a motion sequence containing a single instance of an action.

In these experiments, we use features from 15 joints corresponding to the NiTE Kinect skeleton, 500 hidden neurons in the ELM, the sequence probability model, and 10-fold cross validation. The results are shown in Table 4.3. In [25], a conditional random field threshold model was used to both segment and recognize the actions. We therefore only compare for the substitution errors which are equal to the classification error represented as S in the table. In total, there are 1223 instances in the used data group. The average accuracy of our method is 99.3%, the training time is less than 11 seconds, and the average testing time for a single gesture is 3 milliseconds.

Table 4.3. Gesture recognition results for MSRC-12. G is gesture class; N is the number of instances of each class. S is the number of wrong classified instances in [25].

G	N	S	Our	G	N	S	Our	G	N	S	Our
G1	101	6	0	G5	101	8	6	G9	105	9	0
G2	102	0	0	G6	92	11	0	G10	103	3	1
G3	101	3	1	G7	103	1	1	G11	106	9	0
G4	101	10	0	G8	103	4	0	G12	105	0	0

In [39], the authors analyzed the correctness of the performances given with different kinds of instructions using e.g. random forest classifiers, and concluded that the instructions given in both video and textual form (Video+Text) were superior to the other types of instructions. Therefore we also use the data group based on the Video+Text instructions, which contains 1210 motion sequences. For this data group, we get an overall accuracy of 99.8% which is slightly better than with the video-based instructions. Our results therefore also indirectly confirm the conclusions made in [39].

4.5.2 Berkeley Multimodal Human Action Database

The Berkeley Multimodal Human Action Database (MHAD) [112] contains 11 actions performed by 12 actors, totalling about 660 instances. These actions are full-body actions, such as jump, bend, punch, and throw. Out of the 11 actions, one action is a combination of other two actions: *sitdown* and *standup*. By using the sequence probability model, these two actions are

Table 4.4. Gesture recognition results for MHAD. From Publication IV.

Method (Accuracy)	1-NN[112]	3-NN[112]	K-SVM[112]	Our
Motion capture (%)	74.8	75.6	79.9	99.5
Accelerometer (%)	79.2	81.8	85.4	90.7

misclassified with their combination action *sitdownStandup*. Thus we use the sequence histogram model for this dataset. We train the ELM model without the combination action, feed training data from all the actions into the ELM model and build the histogram model for each action. By using the sequence histogram model, these actions are well distinguished and the average accuracy for these 11 actions reaches 99.5%.

Gesture recognition on accelerometer data

As a multimodal dataset, MHAD contains not only data from a motion capture system but also data from accelerometers. Six accelerometers are attached to the actors, with each sensor providing three-dimensional acceleration values. These values are combined to form a 18-dimensional feature vector, whose elements are then normalized to the range $[-1\ 1]$. We calculate the temporal difference feature vector and concatenate them to form a 36-dimensional feature vector, and use the sequence histogram model similarly as with the mocap data.

The average recognition accuracies are shown in Table 4.4. The table contains also results from three methods using the same dataset from [112]. We can see that for motion capture data we get almost 100% accuracy, and for accelerometer data, our system also reaches above 90% in classification accuracy.

5. Multi-modal gesture recognition

Skeleton data, as the only data modality in a motion capture system, can provide rich information enabling accurate gesture recognition. It also serves well in recognition tasks for RGB-D sensors. Nevertheless, in addition to skeletal data, RGB-D sensors also provide RGB video and raw depth data. In this chapter, we briefly introduce our methodology for multi-modal gesture recognition, which was used in our participation to the ChaLearn 2013 competition [35]. A more specific description of the competition and our participation can be found in Publications VII and VIII.

5.1 Gesture recognition challenges

As the Kinect sensor has become popular among the computer vision research community, many research groups have collected their own Kinect datasets for different purposes in various research topics. Whereas these datasets are often small in scale and not always easily accessible, large public Kinect datasets can be immensely beneficial to the whole research community. Traditionally, many well-established competitions with large databases have been organized for certain important tasks. These public databases often contain comprehensive amounts of data, which require huge efforts to collect but offer a valuable platform to evaluate various methods. There have been several competitions on human action recognition organized in recent years. Before the launch of Kinect, the competitions focused mainly on video data in the surveillance domain. Examples include the contest on semantic description of human activities in 2010 [131] and the VIRAT action recognition challenge in 2011 [113]. More recently, depth data from a single or multiple Kinect devices are supplied along with color or grayscale video from other cameras. For example, in

the Human activities recognition and localization competition two video cameras and one Kinect are used to record the human activities [159].

5.1.1 ChaLearn one-shot-learning gesture challenge

ChaLearn (Challenges in Machine Learning Foundation) has organized gesture recognition contests since 2011. Unlike many other competitions, the challenges collect data only from a single Kinect device, which is closer to the real use cases in daily life. In 2011-2012 ChaLearn organized the One-shot-learning gesture challenge. Over 50,000 hand and arm gestures from 85 different gesture vocabularies were recorded using the Kinect [45]. The RGB and depth videos are provided in the database. The challenge emphasizes learning from very few training examples, as the name of the challenge implies. The gestures are grouped into subtasks based on application domains, such as activities (drinking or writing), body language gestures (crossing your arms) or signals [47]. Each subtask contains 8 to 12 different gestures to be recognized.

5.1.2 ChaLearn multi-modal gesture challenge

In 2013, ChaLearn organized the Multi-modal gesture recognition challenge. This challenge emphasizes user independent gesture recognition from a multi-modal dataset recorded with a Kinect device. The dataset contains RGB video, depth video, user masks, skeleton information, and audio, as shown in Figure 5.1. The RGB and depth video streams are of VGA resolution (640×480) with 8 bits and 11 bits per pixel respectively, and the frame rate is 20 frames per second on average. While the actor performs the gesture, the name of the gesture is also spoken in Italian at the same time and recorded into the audio track. Most of the gestures can be performed either by left or right hand depending on the performer's handedness. The dataset includes 13,858 instances from 20 Italian anthropological gesture categories performed by a total of 27 actors [35].

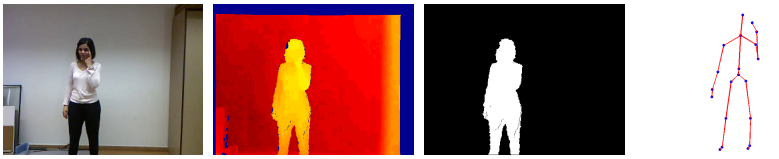


Figure 5.1. Different data modalities in the ChaLearn 2013 multi-modal dataset.

Figure 5.2 depicts key postures for each gesture. The images have been cropped to contain only the main body of the performer for better visualization. It is easy to observe that many gestures are very similar to each other based on only the skeleton data. For example, the gestures (6) and (7) are very similar except for the hand pose; (4) and (18) share the same hand pose and also very similar skeletons. In addition to the interclass similarity, due to the natural and fast hand movements, motion blur easily occurs in the RGB video, as seen Figure 5.3. Consequently, gesture recognition on this dataset using only either RGB video or skeleton data is very challenging.

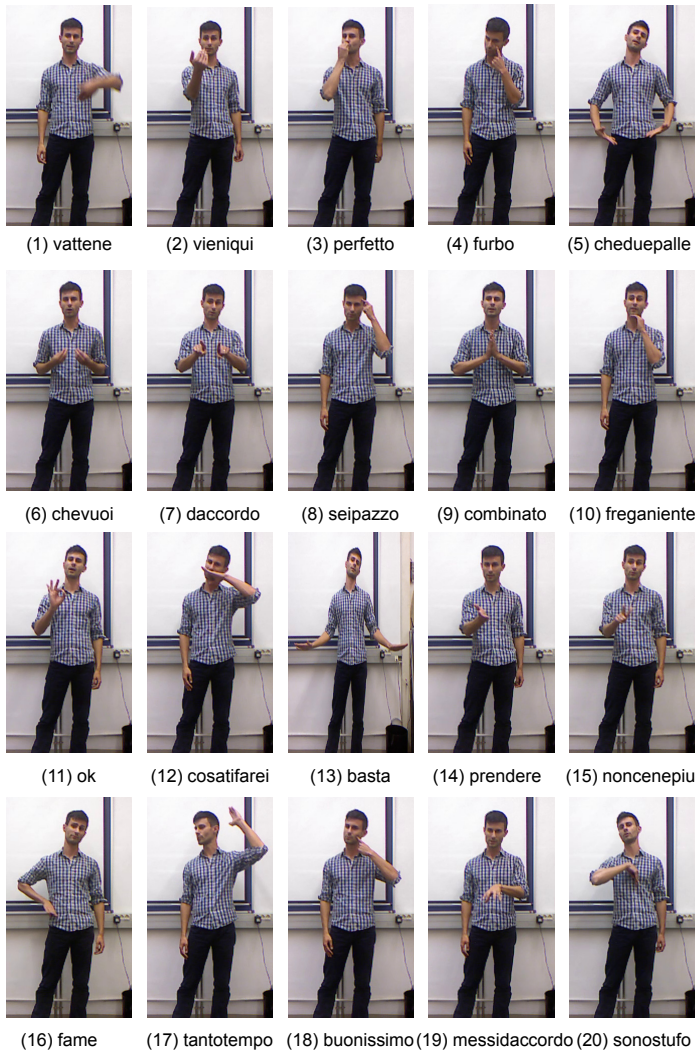


Figure 5.2. ChaLearn 2013 gesture categories.

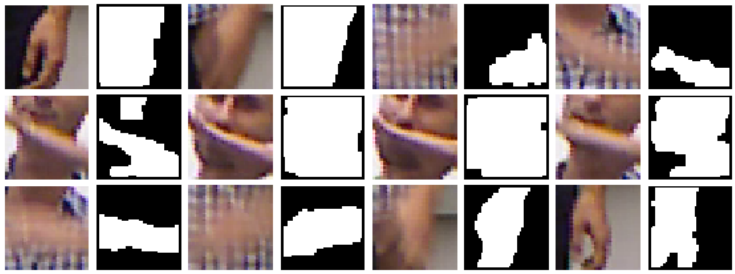


Figure 5.3. A closer look of the left hand image and corresponding depth-based masks from the gesture: costa ti farei (what would you do). Each image is 40×40 pixels. Adapted from Publication VIII ©IEEE.

In the challenge, all top ranked participants use audio data. The winners reported a recognition accuracy of 93.5% by employing only the audio data for the validation set [161]. However, audio data is not commonly available in many daily applications, and people rarely say the name of the activity while they are performing the action naturally. Audio is also highly susceptible to surrounding noise. Therefore in our system we only used the visual modalities shown in Figure 5.1.

5.2 Recognition framework

In our system, we extract several features from the used modalities. Normalized 3D joint positions and pairwise distances between joints are extracted from the skeletal data; HOG, HOG3D, LBP, and Gabor features are extracted from the RGB video, and the HON4D features are extracted from the depth video. Some of these features are concatenated in the early fusion stage to form various combinations. After the early fusion, these features are used to train multiple ELM classifiers on the frame level. The outputs from the ELMs are modeled on the gesture sequence level to generate the gesture predictions. In the late fusion stage, the multiple predictions are aggregated to provide a final classification for a gesture sequence. A diagram of the recognition framework is shown in Figure 5.4.

5.2.1 Skeletal features

In Chapter 4 we showed that skeleton features can be very effective for gesture recognition. We therefore extract the NP and PW features described in Section 4.2.1 from the skeletal data. Figure 5.5 displays one example of NP feature extraction. In this dataset, the cultural sign gestures require

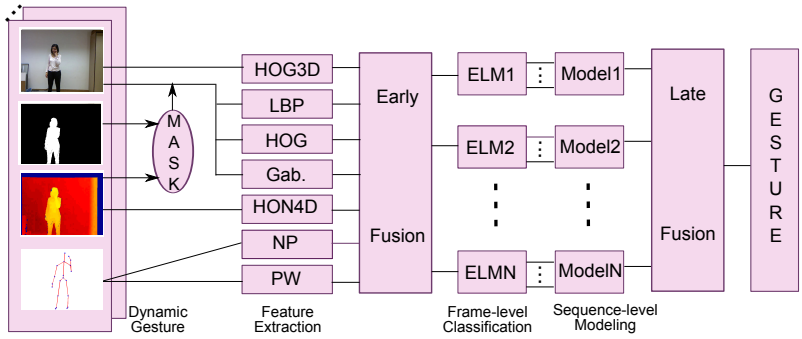
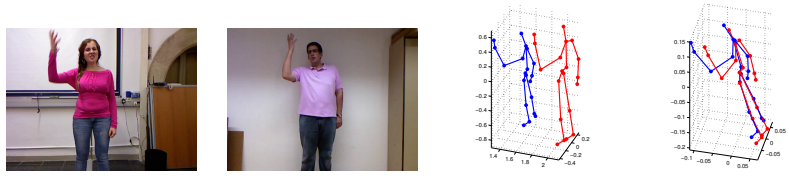


Figure 5.4. The framework of the dynamic gesture recognition system. Adapted from Publication VIII ©IEEE.

only the movement of hands and arms, and hence in the skeletal feature we only use the following upper-body joints: the spine, shoulder center, head, shoulders, elbows, wrists and hands.



(a) **(b)** **(c)** **(d)**

Figure 5.5. RGB frames and the corresponding skeleton information for the same gesture from two different performers. (a) Gesture from A; (b) same gesture from B; (c) original coordinates of A and B;(d) normalized 3D joint position of A and B. From Publication VII.

In the experiments of Section 4.3.1, the PW skeletal feature was found to be a little inferior to the NP feature in overall recognition accuracy. Still, to benefit from the distinctive powers of different features, we utilize both features through early and late fusion to improve the overall accuracy. The PW feature is also extracted for the upper body joints.

5.2.2 Hand features

Skeletal features have been shown to be very effective in many situations [156]. Nevertheless, they are not capable of capturing hand configurations, which often present meaningful linguistic symbols in gestures. Examples shown in Figure 5.2 illustrate one case where the skeletal features are not alone sufficient to distinguish between the gestures (6) and (7). Therefore, the distinction of the hand poses is vital for this recognition task. To capture the differences between hand poses, localizing the hand is a

crucial task. Due to the high dimensionality of the hand configuration, illumination variation, self-occlusion, and motion blur, the tracking of the hands in uncontrolled environments and with natural gestures is very challenging, especially if no initialization is provided.

Skin color is often used to track and segment hands or faces [114]. It can, however, also be difficult in uncontrolled environments. In the ChaLearn datasets, the actions are recorded in several different environments, the lighting is not controlled, and the 27 actors have a big variety of skin colors. Moreover, the hands often occlude each other or the face, and forearms are not covered by clothes in many recordings, which makes it difficult to separate hands from the forearms or the face based only on skin color segmentation. Finally, during the recording performers stand relatively far away from the Kinect device, which causes the hands in the RGB image to occupy rather small areas, roughly bounded by a box of size 40×40 pixels.

The Kinect skeleton model is generated from the depth data, which avoids the difficulties of segmentation based on the RGB data. Therefore, we take advantage of the skeleton model provided in the dataset, use the 2D hand joint pixel coordinates from the skeleton model as the centers of hand locations, and extract features from fixed hand regions around the centers. This way, we obtain the hand locations without any extra computation, but the extracted hand features are affected by the accuracy of the skeleton model. Figure 5.6 shows an example of the hand feature extraction. In this figure, HOG features are extracted from the left and right hand separately. In order to be able to use a shared classifier for both hands, the right hand image is first flipped in the horizontal direction. The size of the hand bounding boxes is 40×40 pixels. The grayscale HOG features use a grid of 2×2 cells and a cell size of 20×20 pixels. To compensate for the inaccuracies of the locations of the hands, we use relatively reasonably large cells of pixels.

The extracted hand regions contain diverse backgrounds. In order to reduce the noise from the background clutter, we try to segment the hand within the hand region from the background. Robust hand segmentation is challenging as discussed above, so we again use the depth information. The dataset provides frame-wise body masks obtained by segmenting the full body of the performer from the background. We use the body mask to segment the hand and other body regions in the hand region from the background. In order to further segment the hand from the body, we use the depth information of the hand bounding box to develop another hand

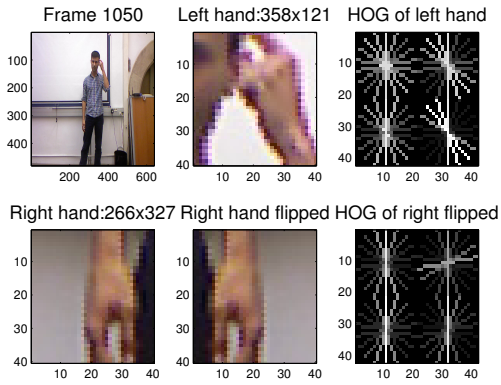


Figure 5.6. HOG features extracted from the left and (flipped) right hand region. From Publication VII.

mask. We assume that the hand is the closest object to the Kinect within the hand bounding box and pixels with a depth value less that of a certain threshold are marked as belonging to the hand. Morphological opening and closing are applied to smooth the hand masks. Figure 5.3 shows the hands and their corresponding hand masks from one gesture instance. To find the most effective hand features, we extract HOG, LBP, Gabor, HOG3D, and HON4D features with different grid and cell sizes for the hand region with and without the masks. Detailed information about the parameters of the used hand features can be found in Publication VII.

5.2.3 Fusion and classification

Each extracted feature has its own advantages and captures distinctive information from the original data. To increase the distinctiveness of the features, we concatenate the features in multiple combinations in early fusion (see Figure 5.4).

Before the ELM training, we also need to consider the handedness of the gesture. In the ChaLearn dataset, most gestures are performed either by only one hand or both hands in a symmetric way. For one-hand gestures, different actors tend to use either left or right hand as dominant based on their handedness. Therefore for each gesture, we determine the dominant hand based on the movement of the arms. Then for each gesture class, separate ELMs are trained for the left and for the right dominant hand respectively. For a testing gesture, we first determine the dominant hand, extract skeleton features and the hand features around the dominant hand

region, and classify with the corresponding ELMs. The outputs from the ELMs are modeled by the sequence probability model as Equation 4.9.

The final stage in the framework is late fusion. After obtaining the sequence-level classification outputs for multiple features, geometric mean is calculated to fuse these outputs, $\bar{d}_m = \prod_j d_m^j$, where d_m^j is the sequence-level classification result for the jth of feature. Finally, we classify a test sequence to the class m with the maximum $\bar{d}_m$ among all classes.

5.2.4 Experiments

The recognition framework employs a modular design regarding to the features and supports any number of parallel features to be used for the recognition. The results for different hand features with multiple grid sizes as single features are shown in Table 5.1. Due to high dimensionality or computational complexity, some features are not extracted.

Table 5.1. Classification accuracies with different hand features. Adapted from Publication VIII ©IEEE.

	static features									temporal features	
	no mask			body mask			hand mask				
grid	HOG	LBP	Gab.	HOG	LBP	Gab.	HOG	LBP	Gab.	HOG3D	HON4D
2×2	59.9	50.9	44.9	62.5	59.3	50.9	68.1	59.5	55.3	54.3	
3×3	64.3	50.8	50.1	65.0	60.7	55.0	**68.9**	59.5	57.6	61.1	63.5*
4×4	65.0	-	49.3	64.7	-	54.8	67.1	-	58.5	60.8	

*) No grid structure used

From Table 5.1 we can observe that HOGs using 3×3 cells seem to be a good compromise of accuracy and feature dimensionality and to be superior to the other features for this data. Hand segmentation is clearly beneficial, which is shown by the higher recognition accuracies with hand masks than without the masks.

A selected set of results using feature fusion is shown in Table 5.2. Similar as in Section 4.3.1, we see that the NP feature for Kinect is slightly superior to PW, but after either early or late fusion of these two skeleton features, the accuracy improves by about 2%. A considerable further improvement can then be obtained by including one or two hand features. With either fusion strategy, this raises the recognition accuracy to about 83–84%. Rather small futher improvements can then be obtained by including even more features. By using several features and both early and late fusion, the system can achieve an overall accuracy of 85.5%. For

comparison, in the ChaLearn 2013 competition the winners' recognition system was based on audio and skeleton data. They use a dynamic time warping based classifier for the skeletons. For the same data set, they report an accuracy of 60.0% based on skeleton features and 93.5% based on audio features [161]. This illustrates the primary role of the audio modalities in the competition.

Table 5.2. Selected fusion results of skeletal and hand features; the symbols "‖" and "+" denote early and late fusion, respectively. The superscript and subscript refer to the used mask and cell structure. Adapted from Publication VIII ©IEEE.

used features	accuracy
NP	71.5
PW	70.4
NP‖PW	73.5
NP+PW	73.1

used features	accuracy
NP+PW+$\mathrm{HOG}^{\mathrm{ha}}_{3\times3}$+$\mathrm{LBP}^{\mathrm{bo}}_{3\times3}$	83.7
NP‖PW‖$\mathrm{HOG}^{\mathrm{ha}}_{3\times3}$	83.9
NP+$\mathrm{HOG}^{\mathrm{ha}}_{3\times3}$+$\mathrm{HOG3D}_{3\times3}$+ (NP‖PW‖$\mathrm{HOG}^{\mathrm{ha}}_{3\times3}$)	**85.5**

In addition to the accuracy, the computational costs such as time and resources also determine whether a system is suitable for a real application. Currently the implementation is written in Matlab, and all experiments were conducted on an Intel(R) Xeon(R) CPU at 3.3 GHz and 16 GB of memory. For example, the feature extraction takes 1.6 ms, 0.026ms, and 24 ms per frame for the NP, PW, and $\mathrm{HOG}^{\mathrm{ha}}_{3\times3}$ features, and classification with a single ELM takes about 0.1 ms per frame. Therefore the recognition with these three features can be easily achieved in real time. Moreover, each ELM takes about 1 to 3 minutes to train for the full ChaLearn 2013 training dataset. Using early fusion of NP, PW, and $\mathrm{HOG}^{\mathrm{ha}}_{3\times3}$, a reasonable trade off between accuracy and complexity, only two ELMs need to be trained. This makes it possible to retrain the system or to learn new gestures with a reasonable delay even for online applications.

6. Summary and discussion

Gesture recognition as a popular research topic, with a couple of decades of intensive efforts, still remains one of the open questions in the field of computer vision. A successful solution to this problem would bring significant influence in many aspects of our daily lives. For example, gestures can be used as commands to control computer programs and robots without specific input devices such as mouses or keyboards. In surveillance, the automatic recognition or even prediction of gestures can reduce dangerous behaviour or even prevent crimes. Sign language recognition can make communication feasible between sign language users and non-users.

In this thesis, we have developed a real-time gesture recognition system for motion capture and RGB-D data with high accuracy for a comparatively large number of gestures. Our simple but robust skeletal features can be a good option among various features, which often require complicated computations. We also extract several appearance-based hand image and depth features. These features improved the recognition accuracy compared to the skeletal features alone, but on the other hand are more costly in terms of computation. Through the experiments, ELM has shown its advantages in both aspects of accuracy and timing compared to several widely-used classifiers.

In our current system, we have assumed that the beginning and end of an action are always known, and that the final classification of the action is given after a completely performed gesture. To achieve this condition, action spotting is required, which is another active research topic. In this thesis, we do not cover this topic but it still has influence on our system. As our system classifies the gestures on the frame level, it is relatively straightforward to predict already during an ongoing gesture. The early

prediction enables more applications for the methodology. This topic of research could definitely be continued further.

Currently, the most mature applications of gesture recognition appear in the game industry, where the game applications require only a limited number of gestures and have a relatively high tolerance for errors. In current HCI and robotics applications, the supported gestures are mostly restricted to about half a dozen of different gestures. However, to make these applications available in real daily life still requires a lot of research to improve the accuracy, robustness and speed of the recognition systems, especially for the huge amount of gestures encountered in everyday life. These difficulties do not only lie on the features and classifiers, but instead the sensors have a crucial role in the recognition. In particular, the Kinect device used in the experiments of this thesis is the first version of its kind. The optimal sensing range is between 1.2m to 3.5m, which significantly restricts the usability in many applications. The RGB image resolution of 640×480 also puts a burden on the image feature extraction. For example, in the ChaLearn multi-modal gesture challenge, the hands only occupy a region of approximately 40×40 pixels. It is very challenging to extract distinctive features from such low resolution images. The skeletal model generated from the depth data is also much noisier and more unstable compared to the motion capture skeleton, which also influences the effectiveness of the skeletal feature.

Compared to face recognition, which is much more widely used in many online applications, a gesture is a dynamic process, which allows a large variation for the same kind of gesture. For example, the action *clapping hand* can be performed in front of the chest, in front of the belly or even with hands raised over the head. There are no standard performances for commonly defined actions. Some actions have only little movement, and these micro-actions often convey hidden information, which is difficult to detect and recognize. For example, during a conversation, a person might slightly and slowly nod her head, which often implies an approval but can be very challenging to detect. Many actions also involve objects, and therefore a meaningful action recognition might also require object recognition. The recognition of the action “holding a gun in hand” definitely requires a different reaction than “holding a mobile in hand”. Such object recognition is also another challenging research topic.

In addition to gesture recognition for motion capture and RGB-D data, we also investigate the matching or alignment of gestures between the skele-

ton models from motion capture and RGB-D data. The proposed method for matching can also be used for gesture retrieval and the evaluation of skeleton models generated by different algorithms from various data sources. Some techniques used for gesture recognition are also shared by our proposed image retrieval system. Based on the image retrieval system, we have built a mobile augmented reality application which can retrieve relative information based on pictures taken by the users. This can be considered as a replacement of current popular quick response (QR) codes which use a matrix barcode. Compared to the QR codes, the recognition of an image from a large database is however much more challenging, and more efforts need to be taken to address this problem.

Bibliography

[1] K. Adistambha, C. Ritz, and I. Burnett. Motion classification using dynamic time warping. In *IEEE 10th Workshop on Multimedia Signal Processing*, 2008.

[2] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff. A unified framework for gesture recognition and spatiotemporal gesture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1685–1699, September 2009.

[3] S. Bailey, C. Powell, S. D. Laycock, and A. M. Day. Interactive exploration of historic information via gesture recognition. In *18th International Conference on Virtual Systems and Multimedia (VSMM)*, pages 211–218. IEEE, 2012.

[4] J. Barbič, A. Safonova, J.-Y. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard. Segmenting motion capture data into distinct behaviors. In *Proceedings of Graphics Interface*, pages 185–194. Canadian Human-Computer Communications Society, 2004.

[5] M. Barnachon, S. Bouakaz, B. Boufama, and E. Guillou. A real-time system for motion retrieval and interpretation. *Pattern Recognition Letters*, 34(15):1789–1798, 2013.

[6] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. In *Proc. ECCV 2006*, May 2006.

[7] I. Bayer and T. Silbermann. A multi modal approach to gesture recognition from audio and video data. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, pages 461–466, New York, NY, USA, 2013.

[8] M. Biao, X. Wensheng, and W. Songlin. A robot control system based on gesture recognition using Kinect. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, 11(5):2605–2611, 2013.

[9] C. M. Bishop et al. *Pattern recognition and machine learning*, volume 1. Springer New York, 2006.

[10] H. Brashear, V. Henderson, K.-H. Park, H. Hamilton, S. Lee, and T. Starner. American sign language recognition in game development for deaf children. In *Proceedings of the 8th international ACM SIGACCESS Conference on Computers and Accessibility*, pages 79–86. ACM, 2006.

[11] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[12] L. Bretzner, I. Laptev, and T. Lindeberg. Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. In *Proceedings of 5th IEEE International Conference on Automatic Face and Gesture Recognition*, pages 423–428. IEEE, 2002.

[13] M. Brown and D. Lowe. Unsupervised 3D object recognition and reconstruction in unordered datasets. In *5th International Conference on 3-D Digital Imaging and Modeling*, pages 56–63. IEEE, 2005.

[14] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2007.

[15] A. E. Bryson. *Applied optimal control: optimization, estimation and control*. CRC Press, 1975.

[16] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983.

[17] L. Cao, R. Ji, Y. Gao, W. Liu, and Q. Tian. Mining spatiotemporal video patterns towards robust action retrieval. *Neurocomputing*, 105:61 – 69, 2013.

[18] J. M. Carmona and J. Climent. A performance evaluation of HMM and DTW for gesture recognition. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 236–243. Springer, 2012.

[19] S. Celebi, A. S. Aydin, T. T. Temiz, and T. Arici. Gesture recognition using skeleton data with weighted dynamic time warping. *Computer Vision Theory and Applications. VISAPP*, 2013.

[20] X. Chai, G. Li, X. Chen, M. Zhou, G. Wu, and H. Li. VisualComm: a tool to support communication between deaf and hearing persons with the Kinect. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, page 76. ACM, 2013.

[21] D. Chen, N.-M. Cheung, S. Tsai, V. Chandrasekhar, G. Takacs, R. Vedantham, R. Grzeszczuk, and B. Girod. Dynamic selection of a feature-rich query frame for mobile video retrieval. In *17th IEEE International Conference on Image Processing (ICIP)*, pages 1017–1020. IEEE, 2010.

[22] D. Chen, S. Tsai, K.-H. Kim, C.-H. Hsu, J. P. Singh, and B. Girod. Low-cost asset tracking using location-aware camera phones. In *SPIE Optical Engineering+ Applications*, pages 77980R–77980R. International Society for Optics and Photonics, 2010.

[23] Q. Chen, N. D. Georganas, and E. M. Petriu. Real-time vision-based hand gesture recognition using Haar-like features. In *Instrumentation and Measurement Technology Conference Proceedings*, pages 1–6. IEEE, 2007.

[24] Z. Cheng, L. Qin, Y. Ye, Q. Huang, and Q. Tian. Human daily action analysis with multi-view and color-depth data. In *Computer Vision–ECCV 2012. Workshops and Demonstrations*, pages 52–61. Springer, 2012.

[25] H. Chung and H.-D. Yang. Conditional random field-based gesture recognition with depth information. *Optical Engineering*, 52(1):017201–017201, 2013.

[26] CMU. Carnegie-Mellon mocap database. *http://mocap.cs.cmu.edu*, 2003.

[27] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[28] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893. IEEE, 2005.

[29] J. G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, 2(7):1160–1169, 1985.

[30] M. de La Gorce, N. Paragios, and D. J. Fleet. Model-based hand tracking with texture, shading and self-occlusions. In *IEEE Conference On Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[31] V. Delaitre, I. Laptev, and J. Sivic. Recognizing human actions in still images: a study of bag-of-features and part-based representations. In *British Machine Vision Conference (BMVC)*, volume 2, page 7, 2010.

[32] L. Deng, H. Leung, N. Gu, and Y. Yang. Automated recognition of sequential patterns in captured motion streams. In *Web-Age Information Management*, pages 250–261. Springer, 2010.

[33] M. Elmezain, A. Al-Hamadi, and B. Michaelis. Hand trajectory-based gesture spotting and recognition using HMM. In *16th IEEE International Conference on Image Processing (ICIP)*, pages 3577–3580. IEEE, 2009.

[34] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108(1):52–73, 2007.

[35] S. Escalera, J. Gonzàlez, X. Baró, M. Reyes, O. Lopes, I. Guyon, V. Athitsos, and H. J. Escalante. Multi-modal gesture recognition challenge 2013: Dataset and results. In *ChaLearn Multi-modal Gesture Recognition Grand Challenge and Workshop, 15th ACM International Conference on Multimodal Interaction*, 2013.

[36] S. R. Fanello, I. Gori, G. Metta, and F. Odone. One-shot learning for real-time action recognition. In *Pattern Recognition and Image Analysis*, pages 31–40. Springer, 2013.

[37] Y. Fang, K. Wang, J. Cheng, and H. Lu. A real-time hand gesture recognition method. In *IEEE International Conference on Multimedia and Expo*, pages 995–998. IEEE, 2007.

[38] L. Fe-Fei, R. Fergus, and P. Perona. A Bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings of 9th IEEE International Conference on Computer Vision*, pages 1134–1141. IEEE, 2003.

[39] S. Fothergill, H. Mentis, P. Kohli, and S. Nowozin. Instructing people for training gestural interactive systems. In *Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems*, pages 1737–1746. ACM, 2012.

[40] W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *International Workshop on Automatic Face and Gesture Recognition*, volume 12, pages 296–301, 1995.

[41] K. Fukunaga and P. M. Narendra. A branch and bound algorithm for computing k-nearest neighbors. *IEEE Transactions on Computers*, 100(7):750–753, 1975.

[42] C. Gao, F. Kong, and J. Tan. Healthaware: Tackling obesity with health aware smart phone systems. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1549–1554. IEEE, 2009.

[43] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *2011 IEEE International Conference on Computer Vision (ICCV)*, pages 415–422. IEEE, 2011.

[44] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, December 2007.

[45] I. Guyon, V. Athitsos, P. Jangyodsuk, H. J. Escalante, and B. Hamner. Results and analysis of the ChaLearn Gesture Challenge 2012. In *Proceedings of 21st International Conference on Pattern Recognition (ICPR)*, Tsukuba, Japan, November 2012.

[46] I. Guyon, V. Athitsos, P. Jangyodsuk, H. J. Escalante, and B. Hamner. Results and analysis of the ChaLearn gesture challenge 2012. In *Advances in Depth Image Analysis and Applications*, pages 186–204. Springer, 2013.

[47] I. Guyon, V. Athitsos, P. Jangyodsuk, B. Hamner, and H. J. Escalante. ChaLearn gesture dataset (CGD2011).

[48] I. Guyon, V. Athitsos, P. Jangyodsuk, B. Hamner, and H. J. Escalante. ChaLearn gesture challenge: Design and first results. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1–6. IEEE, 2012.

[49] H. Hachiya, M. Sugiyama, and N. Ueda. Importance-weighted least-squares probabilistic classifier for covariate shift adaptation with application to human activity recognition. *Neurocomputing*, 80:93–101, 2012.

[50] S. Hadfield and R. Bowden. Hollywood 3D: Recognizing actions in 3D natural scenes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Portland, Oregon, USA, June 2013.

[51] H. Hamer, K. Schindler, E. Koller-Meier, and L. Van Gool. Tracking a hand manipulating an object. In *IEEE 12th International Conference on Computer Vision*, pages 1475–1482. IEEE, 2009.

[52] L. Han, X. Wu, W. Liang, G. Hou, and Y. Jia. Discriminative human action recognition in the learned hierarchical manifold space. *Image and Vision Computing*, 28(5):836–849, 2010.

[53] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[54] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[55] T. K. Ho. Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition*, volume 1, pages 278–282. IEEE, 1995.

[56] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen. Local binary patterns and its application to facial image analysis: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 41(6):765–781, 2011.

[57] G. Huang, H. Zhou, X. Ding, and R. Zhang. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(2):513–529, 2012.

[58] G.-B. Huang, D. H. Wang, and Y. Lan. Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics*, 2(2):107–122, 2011.

[59] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1):489–501, 2006.

[60] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, pages 604–613. ACM, 1998.

[61] B. Ionescu, D. Coquin, P. Lambert, and V. Buzuloiu. Dynamic hand gesture recognition using the skeleton of the hand. *EURASIP Journal on Applied Signal Processing*, 2005:2101–2109, 2005.

[62] Y.-G. Jiang, Z. Li, and S.-F. Chang. Modeling scene and object contexts for human action retrieval with few examples. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(5):674–681, May 2011.

[63] S. Jones and L. Shao. Content-based retrieval of human actions from realistic video databases. *Information Sciences*, 236:56–65, 2013.

[64] V. Kellokumpu, G. Zhao, and M. Pietikäinen. Recognition of human actions using texture descriptors. *Machine Vision and Applications*, 22(5):767–780, 2011.

[65] E. Keogh, T. Palpanas, V. B. Zordan, D. Gunopulos, and M. Cardle. Indexing large human-motion databases. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases*, volume 30 of *VLDB '04*.

[66] C. Keskin, F. Kirac, Y. Kara, and L. Akarun. Real time hand pose estimation using depth sensors. In *IEEE International Conference on Computer Vision Workshops*, pages 1228–1234, Nov 2011.

[67] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *European Conference on Computer Vision (ECCV)*, pages 852–863. Springer, 2012.

[68] W. Z. Khan, Y. Xiang, M. Y. Aalsalem, and Q. Arshad. Mobile phone sensing systems: a survey. *Communications Surveys & Tutorials, IEEE*, 15(1):402–427, 2013.

[69] I.-C. Kim and S.-I. Chien. Analysis of 3D hand trajectory gestures using stroke-based composite hidden Markov models. *Applied Intelligence*, 15(2):131–143, 2001.

[70] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In *British Machine Vision Conference*, pages 995–1004, September 2008.

[71] K. Konolige, M. Agrawal, R. C. Bolles, C. Cowan, M. Fischler, and B. Gerkey. Outdoor mapping and navigation using stereo vision. In *Experimental Robotics*, pages 179–190. Springer, 2008.

[72] L. Kovar and M. Gleicher. Automated extraction and parameterization of motions in large data sets. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 559–568. ACM, 2004.

[73] B. Krüger, J. Tautges, A. Weber, and A. Zinke. Fast local and global similarity searches in large motion capture databases. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 1–10. Eurographics Association, 2010.

[74] A. Kurakin, Z. Zhang, and Z. Liu. A real time system for dynamic hand gesture recognition with a depth sensor. In *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 1975–1979. IEEE, 2012.

[75] R. Labayrade, D. Aubert, and J.-P. Tarel. Real time obstacle detection in stereovision on non flat road geometry through" v-disparity" representation. In *Intelligent Vehicle Symposium*, volume 2, pages 646–651. IEEE, 2002.

[76] K. Lai, J. Konrad, and P. Ishwar. A gesture-driven computer interface using Kinect. In *2012 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, pages 185–188. IEEE, 2012.

[77] B. M. Lake, R. Salakhutdinov, J. Gross, and J. B. Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, pages 2568–2573, 2011.

[78] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

[79] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[80] S.-W. Lee. Automatic gesture recognition for intelligent human-robot interaction. In *7th International Conference on Automatic Face and Gesture Recognition(FGR)*, pages 645–650. IEEE, 2006.

[81] Y.-S. Lee and S.-B. Cho. Activity recognition using hierarchical hidden Markov models on a smartphone with 3D accelerometer. In *Hybrid Artificial Intelligent Systems*, pages 460–467. Springer, 2011.

[82] W. Li, Z. Zhang, and Z. Liu. Expandable data-driven graphical modeling of human actions based on salient postures. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1499–1510, 2008.

[83] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3D points. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 9–14. IEEE, 2010.

[84] Y. Li. Hand gesture recognition using Kinect. In *IEEE 3rd International Conference on Software Engineering and Service Science (ICSESS)*, pages 196–199. IEEE, 2012.

[85] Y. Lin. Efficient motion search in large motion capture databases. *Advances in Visual Computing*, pages 151–160, 2006.

[86] T. Liu, A. W. Moore, K. Yang, and A. G. Gray. An investigation of practical approximate nearest neighbor algorithms. In *Advances in neural information processing systems*, pages 825–832, 2004.

[87] M. Livingston, J. Sebastian, Z. Ai, and J. Decker. Performance measurements for the Microsoft Kinect skeleton. In *2012 IEEE Virtual Reality Workshops (VR)*, pages 119–120. IEEE, 2012.

[88] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the 7th IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157. IEEE, 1999.

[89] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[90] Y. M. Lui. A least squares regression framework on manifolds and its application to gesture recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 13–18, June 2012.

[91] F. Lv and R. Nevatia. Recognition and segmentation of 3-D human action using HMM and multi-class adaboost. In *Computer Vision–ECCV2006*, volume 3954, pages 359–372. Springer, 2006.

[92] U. Mahbub, H. Imtiaz, T. Roy, M. S. Rahman, and M. A. R. Ahad. A template matching approach of one-shot-learning gesture recognition. *Pattern Recognition Letters*, 34(15):1780 – 1788, 2013.

[93] A. Marcos-Ramiro, D. Pizarro-Perez, M. Marron-Romera, L. Nguyen, and D. Gatica-Perez. Body communicative cue extraction for conversational analysis. In *10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–8. IEEE, 2013.

[94] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

[95] A. Menache. *Understanding motion capture for computer animation and video games*. Morgan Kaufmann Publishers, 2000.

[96] M. Meredith and S. Maddock. Motion capture file formats explained. *Department of Computer Science, University of Sheffielf*, 2001.

[97] Microsoft. Kinect for Windows SDK. *http://www.microsoft.com/en-us/kinectforwindows/*.

[98] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[99] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[100] R. Minhas, A. Baradarani, S. Seifzadeh, and Q. Jonathan Wu. Human action recognition using extreme learning machine based on visual vocabularies. *Neurocomputing*, 73(10):1906–1917, 2010.

[101] S. Mitra and T. Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(3):311–324, 2007.

[102] M. Muja and D. Lowe. FLANN-fast library for approximate nearest neighbors user manual, 2009.

[103] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Proceedings of International Conference on Computer Vision Theory and Applications (VISAPP'09)*, Lisboa, Portugal, February 2009.

[104] M. Müller. *Information Retrieval for Music and Motion*. Springer, 2007.

[105] M. Müller, A. Baak, and H.-P. Seidel. Efficient and robust annotation of motion capture data. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 17–26. ACM, 2009.

[106] M. Müller and T. Röder. Motion templates for automatic classification and retrieval of motion capture data. In *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, pages 137–146, Vienna, Austria, 2006.

[107] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber. Documentation mocap database HDM05. Technical Report CG-2007-2, University of Bonn, June 2007.

[108] K. Nandakumar, K. W. Wan, S. M. A. Chan, W. Z. T. Ng, J. G. Wang, and W. Y. Yau. A multi-modal gesture recognition system using audio, video, and skeletal joint data. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, pages 475–482, New York, NY, USA, 2013.

[109] B. Ni, G. Wang, and P. Moulin. RGBD-HuDaAct: A color-depth video database for human daily activity recognition. In *Consumer Depth Cameras for Computer Vision*, pages 193–208. Springer, 2013.

[110] K. Nickel and R. Stiefelhagen. Visual recognition of pointing gestures for human–robot interaction. *Image and Vision Computing*, 25(12):1875–1884, 2007.

[111] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3):299–318, 2008.

[112] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Berkeley MHAD: A comprehensive multimodal human action database. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 53–60, January.

[113] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3153–3160. IEEE, 2011.

[114] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Efficient model-based 3D tracking of hand articulations using Kinect. In *Proceedings of the 22nd British Machine Vision Conference (BMVC)*, 2011.

[115] T. Ojala, M. Pietikäinen, and T. Mäenpää. A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification. In *Proceedings of the Second International Conference on Advances in Pattern Recognition*, ICAPR '01, pages 397–406, London, UK, UK, 2001.

[116] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, July 2002.

[117] E.-J. Ong, H. Cooper, N. Pugeault, and R. Bowden. Sign language recognition using sequential pattern trees. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2200–2207. IEEE, 2012.

[118] OpenNI. Open-source SDK for 3D sensors. *http://www.openni.org/*.

[119] O. Oreifej, Z. Liu, and W. Redmond. HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 716–723. IEEE, 2013.

[120] V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):677–695, 1997.

[121] M. Peris and K. Fukui. Both-hand gesture recognition based on KOMSM with volume subspaces for robot teleoperation. In *2012 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 191–196. IEEE, 2012.

[122] N. Pugeault and R. Bowden. Spelling it out: Real-time ASL fingerspelling recognition. In *2011 IEEE International Conference on Computer Vision Workshops*, pages 1114–1119. IEEE, 2011.

[123] S. Qin, X. Zhu, Y. Yang, and Y. Jiang. Real-time hand gesture recognition from depth images using convex shape decomposition method. *Journal of Signal Processing Systems*, 74(1):47–58, 2014.

[124] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286. IEEE, 1989.

[125] A. Ramey, V. González-Pacheco, and M. A. Salichs. Integration of a low-cost RGB-D sensor in a social robot for gesture recognition. In *Proceedings of the 6th International Conference on Human-robot Interaction*, HRI '11, pages 229–230, New York, NY, USA, 2011. ACM.

[126] C. R. Rao and S. K. Mitra. *Generalized inverse of matrices and its applications*, volume 7. Wiley New York, 1971.

[127] M. Raptis, D. Kirovski, and H. Hoppe. Real-time classification of dance gestures from skeleton animation. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 147–156. ACM, 2011.

[128] Z. Ren, J. Yuan, J. Meng, and Z. Zhang. Robust part-based hand gesture recognition using Kinect sensor. *IEEE transactions on multimedia*, 15(5):1110–1120, 2013.

[129] M. Reyes, G. Dominguez, and S. Escalera. Feature weighting in dynamic time warping for gesture recognition in depth data. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1182–1188. IEEE, 2011.

[130] M. Rodriguez, J. Ahmed, and M. Shah. Action MACH a spatio-temporal maximum average correlation height filter for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2008.

[131] M. S. Ryoo and J. K. Aggarwal. UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html, 2010.

[132] K. Schindler and L. Van Gool. Action snippets: How many frames does human action recognition require? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[133] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, volume 3, pages 32–36. IEEE, 2004.

[134] A. Shimada and R. Taniguchi. Gesture recognition using sparse code of hierarchical SOM. In *19th International Conference on Pattern Recognition (ICPR)*, pages 1–4. IEEE, 2008.

[135] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2011.

[136] L. Sigal, A. O. Balan, and M. J. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(1-2):4–27, 2010.

[137] M. Sjöberg, M. Koskela, S. Ishikawa, and J. Laaksonen. Large-scale visual concept detection with explicit kernel maps and power mean SVM. In *Proceedings of ACM International Conference on Multimedia Retrieval (ICMR2013)*, pages 239–246, Dallas, Texas, USA, April 2013. ACM.

[138] R. Slyper and J. K. Hodgins. Action capture with accelerometers. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 193–199. Eurographics Association, 2008.

[139] Y. Song, D. Demirdjian, and R. Davis. Continuous body and hand gesture recognition for natural human-computer interaction. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(1):5, 2012.

[140] E. Stergiopoulou and N. Papamarkos. Hand gesture recognition using a neural network shape fitting technique. *Engineering Applications of Artificial Intelligence*, 22(8):1141 – 1158, 2009.

[141] E. P. Stuntebeck, J. S. Davis, II, G. D. Abowd, and M. Blount. Healthsense: Classification of health-related sensor data through user-assisted machine learning. In *Proceedings of the 9th Workshop on Mobile Computing Systems and Applications*, HotMobile '08, pages 1–5, New York, NY, USA, 2008.

[142] E. B. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky. Visual hand tracking using nonparametric belief propagation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pages 189–189. IEEE, 2004.

[143] M. Sun, P. Kohli, and J. Shotton. Conditional regression forests for human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3394–3401. IEEE, 2012.

[144] M. Sun, M. Telaprolu, H. Lee, and S. Savarese. An efficient branch-and-bound algorithm for optimal human pose estimation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1616–1623. IEEE, 2012.

[145] J. A. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[146] N. Tanibata, N. Shimada, and Y. Shirai. Extraction of hand features for recognition of sign language words. In *International Conference on Vision Interface*, pages 391–398. Citeseer, 2002.

[147] M. Tenorth, J. Bandouch, and M. Beetz. The TUM kitchen data set of everyday manipulation activities for motion tracking and action recognition. In *IEEE International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS)*, 2009.

[148] S. S. Tsai, D. Chen, J. P. Singh, and B. Girod. Rate-efficient, real-time CD cover recognition on a camera-phone. In *Proceedings of the 16th ACM International Conference on Multimedia*, pages 1023–1024. ACM, 2008.

[149] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, Nov 2008.

[150] M. van Heeswijk, Y. Miche, E. Oja, and A. Lendasse. GPU-accelerated and parallelized ELM ensembles for large-scale regression. *Neurocomputing*, 74(16):2430–2437, 2011.

[151] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3539–3546. IEEE, 2010.

[152] A. Vieira, T. Lewiner, W. Schwartz, and M. Campos. Distance matrices as invariant features for classifying MoCap data. In *21st International Conference on Pattern Recognition (ICPR)*, Tsukuba, Japan, 2012.

[153] C. Von Hardenberg and F. Bérard. Bare-hand human-computer interaction. In *Proceedings of the 2001 Workshop on Perceptive User Interfaces*, pages 1–8. ACM, 2001.

[154] J. Wan, Q. Ruan, W. Li, and S. Deng. One-shot learning gesture recognition from RGB-D data using bag of features. *Journal of Machine Learning Research*, 14(1):2549–2582, January 2013.

[155] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. Robust 3D action recognition with random occupancy patterns. In *Proceedings of the 12th European Conference on Computer Vision*, ECCV'12, pages 872–885. Springer, 2012.

[156] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1290–1297. IEEE, 2012.

[157] J.-Y. Wang and H.-M. Lee. Recognition of human actions using motion capture data and support vector machine. In *WRI World Congress on Software Engineering, WCSE'09*, volume 1, pages 234–238. IEEE, 2009.

[158] J. Webb and J. Ashley. *Beginning Kinect Programming with the Microsoft Kinect SDK*. Apress, 2012.

[159] C. Wolf, J. Mille, L. Lombardi, O. Celiktutan, M. Jiu, M. Baccouche, E. Dellandréa, C.-E. Bichot, C. Garcia, and B. Sankur. The LIRIS human activities dataset and the ICPR 2012 human activities recognition and localization competition. Technical Report RR-LIRIS-2012-004, LIRIS Laboratory, 2012.

[160] J. Wu. Power mean SVM for large scale visual classification. In *Proceedings of The IEEE Int'l Conference on Computer Vision and Pattern Recognition (CVPR 2012)*, Providence, USA, June 2012.

[161] J. Wu, J. Cheng, C. Zhao, and H. Lu. Fusing multi-modal features for gesture recognition. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, pages 453–460, New York, NY, USA, 2013.

[162] Y. Wu and T. S. Huang. View-independent recognition of hand postures. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 88–94. IEEE, 2000.

[163] L. Xia and J. Aggarwal. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2834–2841. IEEE, 2013.

[164] L. Xia, C.-C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3D joints. In *CVPR Workshops*, pages 20–27. IEEE, 2012.

[165] A. Y. Yang, S. Iyengar, S. Sastry, R. Bajcsy, P. Kuryloski, and R. Jafari. Distributed segmentation and classification of human actions using a wearable motion sensor network. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8. IEEE, 2008.

[166] H.-D. Yang, A.-Y. Park, and S.-W. Lee. Gesture spotting and recognition for human–robot interaction. *IEEE Transactions on Robotics*, 23(2):256–270, 2007.

[167] M.-H. Yang, N. Ahuja, and M. Tabb. Extraction of 2D motion trajectories and its application to hand gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1061–1074, 2002.

[168] X. Yang and Y. Tian. EigenJoints-based action recognition using Naive-Bayes-Nearest-Neighbor. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 14–19. IEEE, 2012.

[169] X. Yang, C. Zhang, and Y. Tian. Recognizing actions using depth motion maps-based histograms of oriented gradients. In *Proceedings of the 20th ACM International Conference on Multimedia*, pages 1057–1060. ACM, 2012.

[170] Y. Yin and R. Davis. Gesture spotting and recognition using salience detection and concatenated hidden Markov models. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, ICMI '13, pages 489–494, New York, NY, USA, 2013. ACM.

[171] H.-S. Yoon, J. Soh, Y. J. Bae, and H. Seung Yang. Hand gesture recognition using combined features of location, angle and velocity. *Pattern Recognition*, 34(7):1491–1501, 2001.

[172] X. Zabulis, H. Baltzakis, and A. Argyros. Vision-based hand gesture recognition for human-computer interaction. *The Universal Access Handbook. LEA*, pages 34.1–34.30, 2009.

[173] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti. American sign language recognition with the Kinect. In *Proceedings of the 13th International Conference on Multimodal Interfaces*, pages 279–286. ACM, 2011.

[174] M. Zanfir, M. Leordeanu, and C. Sminchisescu. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2752–2759. IEEE, 2013.

[175] X. Zhao, X. Li, C. Pang, and S. Wang. Human action recognition based on semi-supervised discriminant analysis with global constraint. *Neurocomputing*, 105:45 – 50, 2013.

[176] F. Zhou, F. Torre, and J. K. Hodgins. Aligned cluster analysis for temporal segmentation of human motion. In *8th IEEE International Conference on Automatic Face & Gesture Recognition*, pages 1–7. IEEE, 2008.

[177] Y. Zhu, W. Chen, and G. Guo. Fusing spatiotemporal features and joints for 3D action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 486–491. IEEE, 2013.

DISSERTATIONS IN INFORMATION AND COMPUTER SCIENCE

Aalto-DD127/2014 Zhang, He
Advances in Nonnegative Matrix Decomposition with Application to Cluster Analysis. 2014.

Aalto-DD138/2014 Sovilj, Dušan
Learning Methods for Variable Selection and Time Series Prediction. 2014.

Aalto-DD144/2014 Eirola, Emil
Machine learning methods for incomplete data and variable selection. 2014.

Aalto-DD149/2014 Äijö, Tarmo
Computational Methods for Analysis of Dynamic Transcriptome and Its Regulation Through Chromatin Remodeling and Intracellular Signaling. 2014.

Aalto-DD156/2014 Sjöberg, Mats
From pixels to semantics: visual concept detection and its applications. 2014.

Aalto-DD157/2014 Adhikari, Prem Raj
Probabilistic Modelling of Multiresolution Biological Data. 2014.

Aalto-DD171/2014 Suvitaival, Tommi
Bayesian Multi-Way Models for Data Translation in Computational Biology. 2014.

Aalto-DD177/2014 Laitinen, Tero
Extending SAT Solver with Parity Reasoning. 2014.

Aalto-DD178/2014 Gonçalves, Nicolau
Advances in Analysis and Exploration in Medical Imaging. 2014.

Aalto-DD191/2014 Kindermann, Roland
SMT-based Verification of Timed Systems and Software. 2014.

Have you ever wondered how many actions you perform each day? Have you ever wondered how computers or robots can be operated by hand gestures without touching or any physical input devices? Have you been amazed by computers recognizing sign language?

These questions are all related to a wide research area called action recognition. In this thesis, methods to answer such questions are provided, from the basic principles to the state-of-the-art research in this area, from theories to applications, from simple explanations to complex solutions. In particular, the developed solutions combine a variety of features from multimodal data with extreme learning machine classifiers. The resulting system can recognize large numbers of different actions in real time with high accuracy.

Aalto-DD 207/2014

ISBN 978-952-60-6013-2 (printed)
ISBN 978-952-60-6014-9 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Information and Computer Science
www.aalto.fi

BUSINESS + ECONOMY

ART + DESIGN + ARCHITECTURE

SCIENCE + TECHNOLOGY

CROSSOVER

DOCTORAL DISSERTATIONS